

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

**Možnosti pokročilé práce s Google
Sheets z prostředí MATLAB a Scilab**

**Possibilities of Advanced Work with
Google Sheets Using MATLAB and
Scilab Software Tools**

Zadání bakalářské práce

Student:

Jakub Malaník

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2601R013 Telekomunikační technika

Téma:

Možnosti pokročilé práce s Google Sheets z prostředí MATLAB a Scilab
Possibilities of Advanced Work with Google Sheets Using MATLAB and Scilab Software Tools

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je popsat kromě základních možností práce s daty v Google Sheets z prostředí MATLAB a Scilab také pokročilé možnosti práce, jako jsou nastavování vlastností jednotlivým buňkám, kopírování listů v rámci jednoho souboru apod.

1. Popište možnosti práce s Google Sheets ve webovém prohlížeči a s použitím software pro synchronizaci dat mezi lokálním úložištěm (HDD) a úložištěm Google Drive.
2. Popište základní možnosti načítání dat z Google Sheets do prostředí MATLAB a Scilab a naopak export dat z prostředí MATLAB a Scilab do Google Sheets. Zaměřte se na možnosti práce s tabulkami, které obsahují proměnné různých typů (string, double, int16 apod.).
3. Popište po konzultaci s vedoucím práce některé pokročilé možnosti práce s Google Sheets z prostředí MATLAB a Scilab, jako jsou vyčítání vlastností jednotlivých buněk (barva buňky, barva písma, zarovnání apod.), a také nastavování parametrů buňkám či blokům buněk (ohraničení tabulky apod.).
4. Popište možnosti práce s celými listy v Google Sheets - vytvoření nového listu, kopírování listu, přejmenování listu, smazání listu apod.
5. Proveďte rešerši možností práce s grafy v Google Sheets z prostředí MATLAB a Scilab.
6. Zhodnoťte možnosti práce s Google Sheets z prostředí MATLAB a Scilab s ohledem na možnosti práce off-line a následnou synchronizaci dat. Zhodnoťte také možnosti napříč jednotlivými platformami operačních systémů (MS Windows, Linux, Mac).

Seznam doporučené odborné literatury:

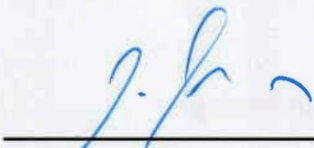
- [1] ALTMAN, Yair M. *Undocumented secrets of MATLAB-Java programming*. Boca Raton: CRC Press, c2012, xxi, 663 s., [16] s. obr. příl. ISBN 978-1-4398-6903-1.
- [2] ALTMAN, Yair M. *Accelerating MATLAB® performance: 1001 tips to speed up MATLAB programs*. Boca Raton: CRC Press, 2015. ISBN 978-1-4822-1129-0.
- [3] URL: <http://undocumentedmatlab.com/>

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Skapa, Ph.D.**

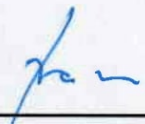
Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2019



prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry





prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 26. dubna 2019


.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.

V Ostravě 26. dubna 2019


.....

Rád bych tímto poděkoval svému vedoucímu bakalářské práce Ing. Janu Skapovi, Ph.D. za odbornou pomoc, trpělivost a ochotu v době psaní mé bakalářské práce.

Abstrakt

Cílem práce je popsat kromě základních možností práce s daty v Google Sheets z prostředí MATLAB a Scilab také pokročilé možnosti práce. Základními možnostmi je myšleno import a export hodnot z/do Google Sheets. Pokročilé možnosti jsou výpis hodnot a vlastností, editace vlastností, práce s grafy a listy.

Klíčová slova: Google Sheets, MATLAB, Scilab

Abstract

The aim of the thesis is to describe in addition to the basic possibilities of working with data in Google Sheets from MATLAB and Scilab also advanced work possibilities. The basic options are to import and export values from/to Google Sheets. Advanced options are listing values and properties, editing properties, working with charts and sheets.

Key Words: Google Sheets, MATLAB, Scilab

Obsah

Seznam použitých zkratk a symbolů	10
Seznam obrázků	11
1 Úvod	13
2 Možnosti synchronizace dat mezi lokálním úložištěm (HDD) a úložištěm Google Drive	14
2.1 Popis Google Drive	14
2.2 Synchronizace v OS Windows 10 s Backup and Sync	14
2.3 Synchronizace v OS Linux Ubuntu 18.04 s Gnome Online Accounts	16
2.4 Synchronizace v OS Linux Ubuntu 18.04 s Google Drive Ocamlfuse	18
2.5 Formát ukládání Google Sheets tabulek na HDD	19
3 Google Sheets	20
3.1 Google Sheets API v4	20
4 Vyčítání hodnot z Google sheets prostřednictvím Matlabu	22
4.1 Vygenerování API klíče	22
4.2 Nasdílení tabulky	25
4.3 Vyčítání hodnot tabulky	26
5 Vyčítání hodnot a vlastností z Google sheets prostřednictvím Matlabu	28
5.1 Vyčítání určitých vlastností	31
6 Nahrávání dat do Google Sheets prostřednictvím Matlabu	36
6.1 Vytvoření Client ID a Client Secret	36
6.2 Vygenerování přístupových tokenů	37
6.3 Nahrávání hodnot	37
6.4 Změna barvy pozadí	39
6.5 Změna barvy textu	40
6.6 Změna formátu textu	41
6.7 Editace ohraničení	44
6.8 Přidávání grafů	48
7 Operace s listy Google Sheets prostřednictvím Matlabu	52
7.1 Praktické ukázky	52

8	Práce s Google Sheets prostřednictvím Scilabu	54
8.1	Vyčítání hodnot	54
8.2	Vyčítání hodnot a vlasností	55
8.3	Zápis hodnot	56
8.4	Změna barvy pozadí	57
8.5	Změna barvy textu	58
8.6	Změna formátu	59
8.7	Editace ohraničení	60
8.8	Přidávání grafů	61
8.9	Práce s listy	62
9	Zhodnocení	63
	Literatura	64

Seznam použitých zkratek a symbolů

MATLAB	– Matrix Laboratory
PC	– Personal Computer
FUSE-base	– Filesystem in Userspace
PPA	– Personal Package Archives
URL	– Uniform Resource Locator
API	– Application Programming Interface

Seznam obrázků

1	Instalace Backup and Sync	15
2	Synchronizace Backup and Sync	15
3	Výběr Gnome	16
4	Přihlášení Gnome	16
5	Povolení Gnome	17
6	Výsledek Gnome	17
7	Spojení Ocamlfuse	19
8	Vytvoření Projektu	22
9	Vytvoření Projektu 2	22
10	Povolení API	22
11	Hledání Google Sheets API	23
12	Povolení API 2	23
13	Vytvoření API klíče	23
14	Vytvoření API klíče 2	24
15	Vytvoření API klíče 3	24
16	Sdílení tabulky	25
17	Sdílení tabulky 2	25
18	Sdílení tabulky 3	25
19	Tabulka hodnoty	27
20	Tabulka vyčítání vlastností	34
21	Vygenerování Client Secret a Client ID	36
22	Vygenerování Client Secret a Client ID 2	36
23	Tabulka zápis hodnot	38
24	Tabulka souřadnice	39
25	Tabulka změna barvy pozadí	40
26	Tabulka změna barvy textu	41
27	Tabulka změna formátu 1	43
28	Tabulka formátu 2	43
29	Tabulka ohraničení 1	45
30	Tabulka ohraničení 2	46
31	Tabulka ohraničení 3	47
32	Tabulka ohraničení 4	47
33	Tabulka změna hodnot	49
34	Tabulka grafy 1	50
35	Tabulka grafy 2	51
36	Zápis hodnot Scilab	56
37	Změna barvy pozadí Scilab	57

38	Změna barvy textu Scilab	58
39	Editace formátu Scilab	59
40	Editace ohraničení Scilab	60
41	Přidání grafů Scilab	61

1 Úvod

Cílem této práce je popsat základní a pokročilé možnosti práce s daty v Google sheets z prostředí MATLAB a Scilab. Dokument je rozdělen do několika částí. V druhé kapitole popisují možnosti synchronizace dat mezi lokálním uložištěm a uložištěm Google Drive na platformách Windows 10 a Linux Ubuntu 18.04. V dalších třech kapitolách se zabývám základními a pokročilými možnostmi čtení dat z Google Sheets do prostředí MATLAB. Šestá kapitola se zaměřuje na základní a pokročilé možnosti zápisu dat do Google Sheets z prostředí MATLAB. Kapitola sedmá popisuje práci s celými listy. A v polední kapitole se zabývám prací s Google Sheets prostřednictvím Scilabu. K vytvoření této práce byl použit MATLAB 2017b 64-bit a Scilab 5.5.2 64-bit.

2 Možnosti synchronizace dat mezi lokálním úložištěm (HDD) a úložištěm Google Drive

2.1 Popis Google Drive

Google Drive je bezplatná služba pro ukládání dat v cloudu, která umožňuje uživatelům ukládat a přistupovat k souborům online. Služba synchronizuje uložené dokumenty, fotografie a další na všech zařízeních uživatele, včetně mobilních zařízení, tabletů a počítačů.

Google Drive je synchronizován s dalšími službami a systémy společnosti – včetně Google Docs, Gmail, Chrome, YouTube, Google Analytics a Google+.

2.2 Synchronizace v OS Windows 10 s Backup and Sync

Google Backup and Sync je bezplatná aplikace pro MAC a PC, která slouží pro bezpečnou zálohu dat na úložiště Google Drive a Google Photos. Backup and Sync vychází již z existujícího klienta Google Drive Sync s novým uživatelským rozhraním a několika novými funkcemi.

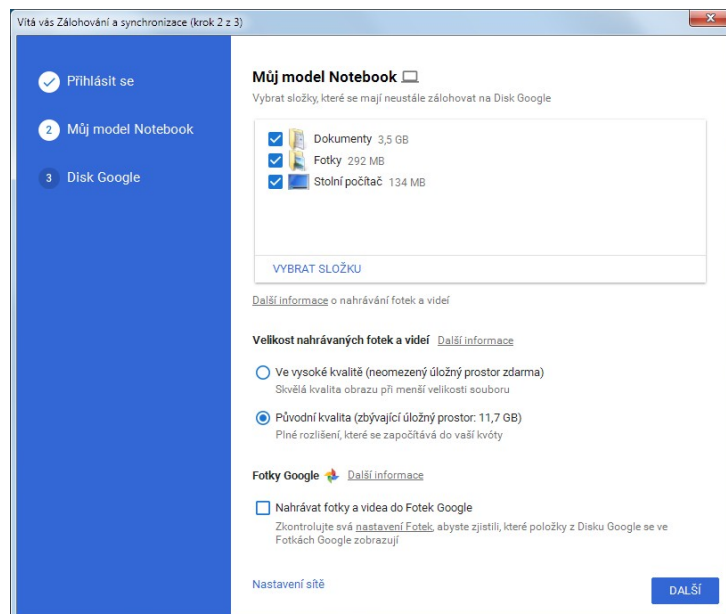
2.2.1 Stažení programu

Nejprve je nutné stáhnout instalaci z googlu <https://www.google.com/drive/download>. Po spuštění instalace jsou staženy instalační soubory a následně je spuštěna instalace.

2.2.2 Instalace a synchronizace programu

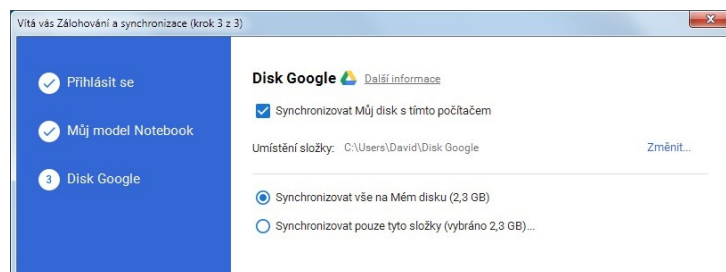
Zobrazí se uvítací obrazovka, kde kliknem na 'Začínáme'. Objeví se okno, kde je nutné se nejprve přihlásit k Google účtu, který chceme synchronizovat s PC.

V dalším kroku si nastavíme složky, které chceme zálohovat do prostředí Google Drive. Je-li aplikace řeší zálohování fotografií a videa, je potřeba nastavit, zda-li se budou fotografie a videa zálohovat na Google Photos nebo přímo na Google Drive. Také se musíme rozhodnout, jestli fotografie a videa budou nahrávány v původní(omezený úložný prostor) nebo vysoké kvalitě(neomezený úložný prostor).



Obrázek 1: Instalace Backup and Sync

Poslední krok instalace je synchronizace. To znamená, že s těmito daty budeme moci pracovat přímo z Průzkumníka/Správce souborů. V počítači bude vytvořena složka Disk Google, do které budou staženy soubory, které jsme si zvolili při synchronizaci. Pokud do této složky něco přidáme, automaticky se nahraje online.



Obrázek 2: Synchronizace Backup and Sync

Ihned po nastavení se začnou data odesílat do Google Disku. Pokud se přihlásíme v prohlížeči do Google Disku, v položce 'Počítač' uvidíme Náš počítač a jednotlivé složky, které jsme pro zálohování vybrali.

2.3 Synchronizace v OS Linux Ubuntu 18.04 s Gnome Online Accounts

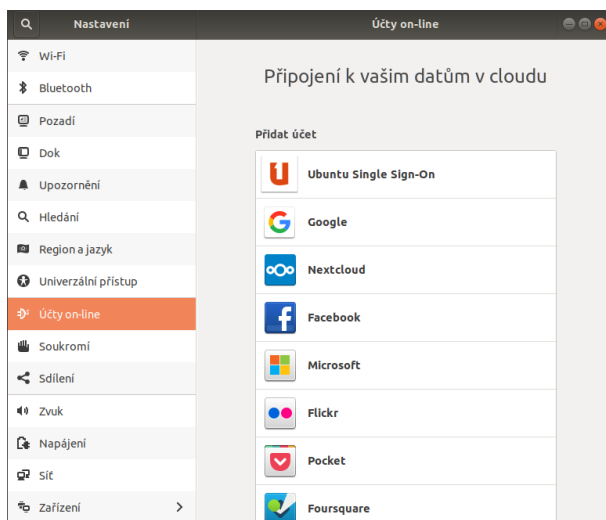
Gnome je uživatelské rozhraní umožňující synchronizaci dat s Google Drive prostřednictvím Online Accounts nacházející se v systémovém nastavení Linuxu. V první řadě je potřeba nainstalovat Gnome Online Accounts přes terminál následujícím příkazem:

```
sudo apt install gnome-online-accounts
```

Po instalaci spustíme Gnome Online Accounts:

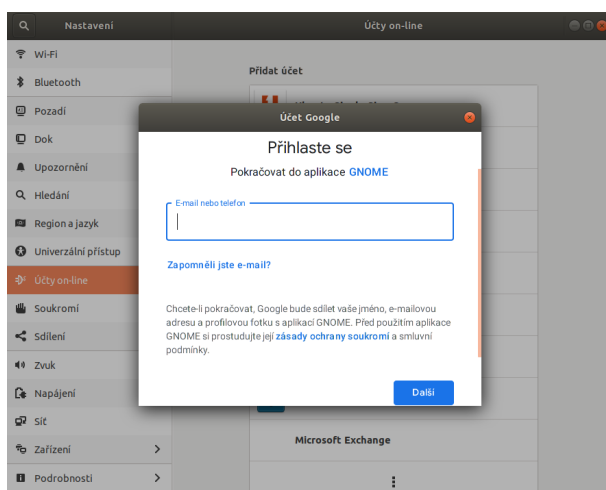
```
gnome-control-center online-accounts
```

Vybereme Google account:



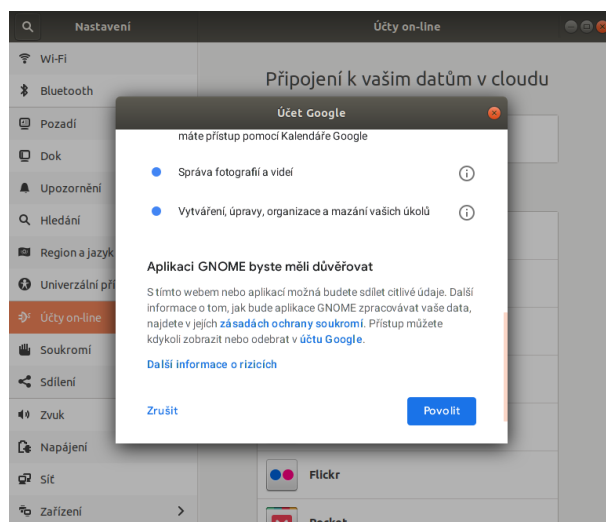
Obrázek 3: Výběr Gnome

Následně zadáme Google účet a poté heslo:



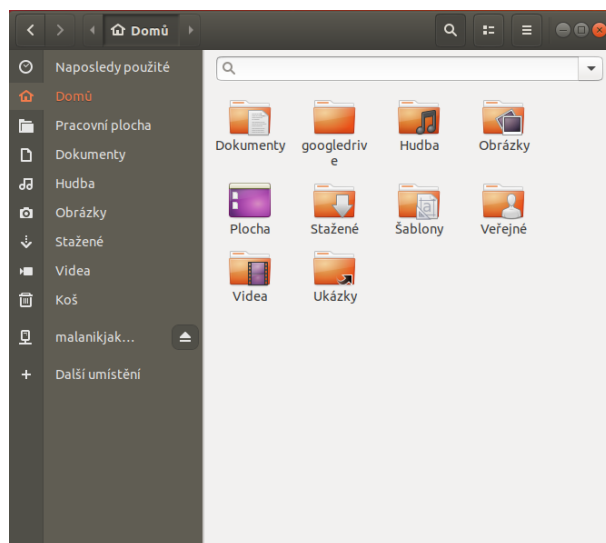
Obrázek 4: Přihlášení Gnome

Povolíme přístup k Google účtu:



Obrázek 5: Povolení Gnome

A nyní je náš účet zesynchronizován s Gnome Online Accounts. V tuhle chvíli můžeme přistupovat k našim souborům na Google Drive prostřednictvím File Manageru v Linux systému:



Obrázek 6: Výsledek Gnome

2.4 Synchronizace v OS Linux Ubuntu 18.04 s Google Drive Ocamlfuse

Google drive Ocamlfuse je FUSE-based souborový systém pro Google Drive, který je možno používat k připojení Google Drive v Linuxu. K jeho funkcím patří:

- plná podpora pro čtení a zápis souborů a složek
- podpora více účtů
- podpora manipulací s duplikovanými soubory(i přístup ke koši Google Drive)
- podpora přístupu (jen pro čtení) k souborům Google Docs (formát ODT, ODS a PDF)

2.4.1 Instalace Google Drive Ocamlfuse

Google Drive Ocamlfuse budeme instalovat přes terminál

2.4.1.1 Instalace PPA

Nejprve je nutné nainstalovat balíček:

```
sudo add-apt-repository ppa:alessandro-strada/ppa
```

2.4.1.2 Instalace Ocamlfuse

Následně je možné nainstalovat Ocamlfuse:

```
sudo apt install google-drive-ocamlfuse
```

2.4.1.3 Propojení Ocamlfuse s Linuxem

Jakmile je aplikace nainstalovaná, můžeme ji spustit:

```
google-drive-ocamlfuse
```

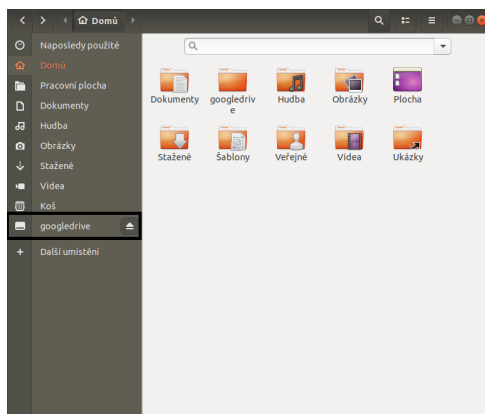
Po spuštění aplikace se otevře prohlížeč, kde bude aplikace požadovat přístup k našemu Google Drive účtu. Následně se přihlásíme a povolíme přístup našim souborům. Další krok je vytvoření složky, to můžeme také provést v terminálu:

```
mkdir ~/googledrive
```

Tímto vytvoříme složku googledrive v našem domovském adresáři. K připojení nově vytvořené složky k Google Drive použijeme následující příkaz:

```
google-drive-ocamlfuse ~/googledrive
```

Pokud teď otevřeme průzkumník, vidíme složku googledrive jako fyzicky připojený disk.



Obrázek 7: Spojení Ocamlfuse

2.5 Formát ukládání Google Sheets tabulek na HDD

Program	OS	Formát
Backup and Sync	Windows 10	.vnd.google-apps.spreadsheets
Gnome Online Accounts N	Linux Ubuntu 18.04	.vnd.oasis.opendocument.spreadsheets (.ods)
Google Drive Ocamlfuse	Linux Ubuntu 18.04	.gsheets

3 Google Sheets

Google Sheet je online tabulková aplikace umožňující uživateli vytvářet, formátovat a sdílet tabulky s ostatními. Je to webová aplikace, jenž je součástí Google Drive a je spustitelná přes prohlížeče Chrome, Firefox, Internet Explorer 11, Microsoft Edge a Safari. Mobilní aplikace je také dostupná na systém Android a iOS. Podporuje následující tabulkové formáty .xlsx, .xls, .xlsm, .xlt, xltx, .ods, .csv, .xtx, .tsv, .tab

Google Sheets umožňuje:

- vytváření nebo nahrání již existující tabulky
- úpravu a formátování tabulky
- používání vzorců a výpočtů
- používání grafů pro zobrazení informací z tabulek
- vytváření formulářů
- export tabulek do různých formátů

Výhody používání Google Sheets:

- možnost práce s tabulkou z více zařízení - dokument je uložen na cloudu (Google Drive)
- sdílení tabulky s ostatními – real-time editování, komentování a chatování
- možnost využívat další Google produkty jako například Google Forms, Google Translate nebo Google Finance

3.1 Google Sheets API v4

Google Sheets API umožňuje uživateli číst a modifikovat všechny aspekty tabulek prostřednictvím kódu. API nabízí 2 hlavní možnosti práce s Google Sheets:

- čtení/zápis hodnot buněk
- čtení/zápis všech aspektů tabulek

O možnostech zápisu a čtení z tabulek se budeme bavit dále v práci.

3.1.1 Běžné výrazy

V této sekci budou popsány některé důležité termíny, se kterými se budeme potýkat v průběhu práce.

3.1.1.1 Spreadsheet ID

Všechny metody používané v Sheets API vyžadují Spreadsheet ID. Tento parametr je využit k identifikaci tabulky, ke které budeme přistupovat nebo ji měnit. Je to hodnota, kterou lze vyčíst z URL dané tabulky:

```
https://docs.google.com/spreadsheets/d/spreadsheetId/edit#gid=0
```

3.1.1.2 Sheet ID

Každý list tabulky má název a ID. Sheet ID je často využíváno v Sheets API k upřesnění, jaký list tabulky je čten nebo upravován. V URL dané tabulky lze Sheet ID vyčíst jako parametr 'gid':

```
https://docs.google.com/spreadsheets/d/spreadsheetId/edit#gid=sheetId
```

3.1.1.3 A1 notace

Některé API metody vyžadují zadání rozsahu v A1 notaci. Je to string, který vypadá například takto: 'List1!A1:B2', kde 'List1' je název listu tabulky 'A1:B2' je rozsah buněk, na který odkazujeme.

Příklady jednotlivých rozsahů:

- List1!A1:B2 – odkazuje na první dvě buňky ve vrchních dvou řádcích v List1
- List1!A:A – odkazuje na všechny buňky v prvním sloupci v List1
- List1!1:2 – odkazuje všechny buňky v prvních dvou sloupcích v List1
- List1!A5:A – odkazuje na všechny buňky prvního sloupce od pátého řádku a dál v List1
- A1:B2 – odkazuje na první dvě buňky ve vrchních dvou řádcích prvního viditelného listu
- List1 – odkazuje na všechny buňky v List1

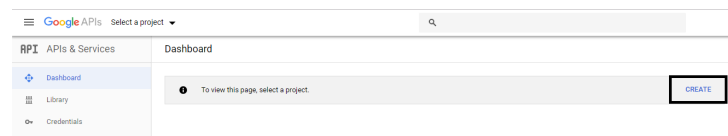
4 Vyčítání hodnot z Google sheets prostřednictvím Matlabu

4.1 Vygenerování API klíče

Před samotným vyčítáním hodnot z tabulky je potřeba vygenerování API klíče. API klíč je jednoduchý string, který nám umožní přistupovat k tabulce. Klíč vygenerujeme následujícím způsobem:

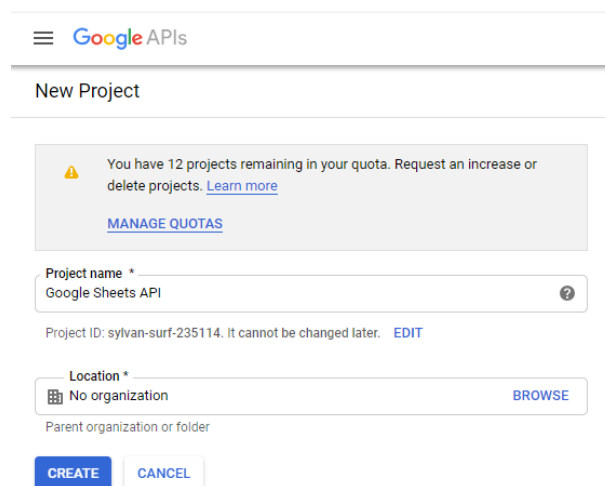
V první řadě se přihlásíme v prohlížeči na svůj Google účet.

Na linku: <https://console.developers.google.com> budeme souhlasit s podmínkami používání API a následně vytvoříme projekt:



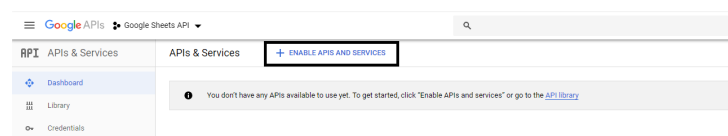
Obrázek 8: Vytvoření Projektu

Do kolonky Project name napíšeme jméno projektu, nejčastěji se volí jméno podle toho, na co chceme projekt použít. Poté klikneme na 'CREATE', tím vytvoříme projekt.



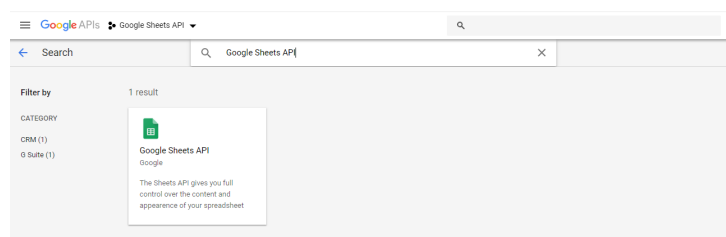
Obrázek 9: Vytvoření Projektu 2

Když máme vytvořený projekt, můžeme přejít na 'ENABLE APIS AND SERVICES'.



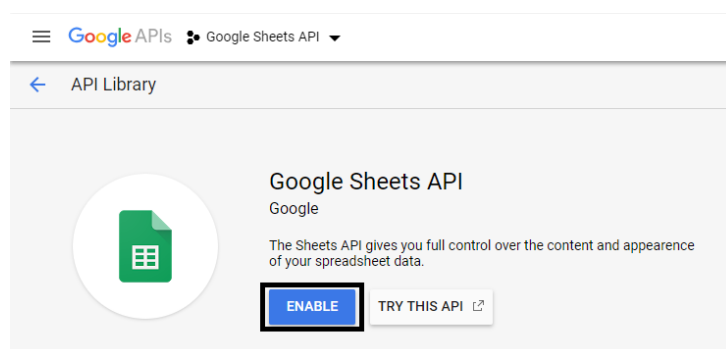
Obrázek 10: Povolení API

V seznamu najdeme Google Sheets API a klikneme na ni.



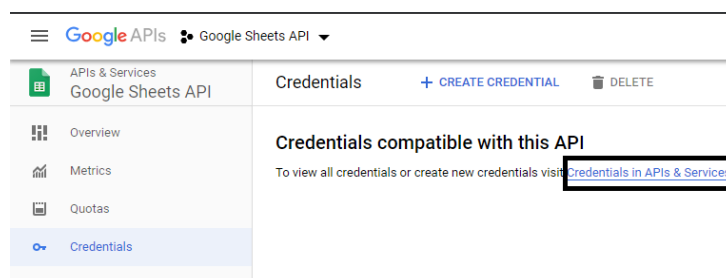
Obrázek 11: Hledání Google Sheets API

Povolíme Google Sheets API



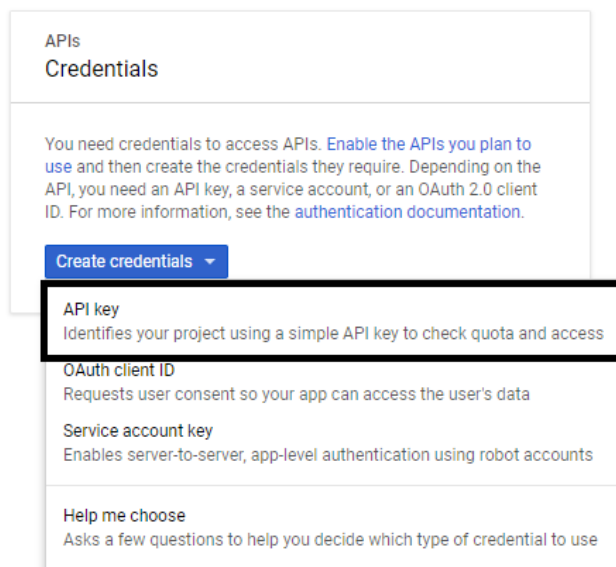
Obrázek 12: Povolení API 2

V levém menu přejdeme na Credentials a klikneme na 'Credentials in APIs & Services'



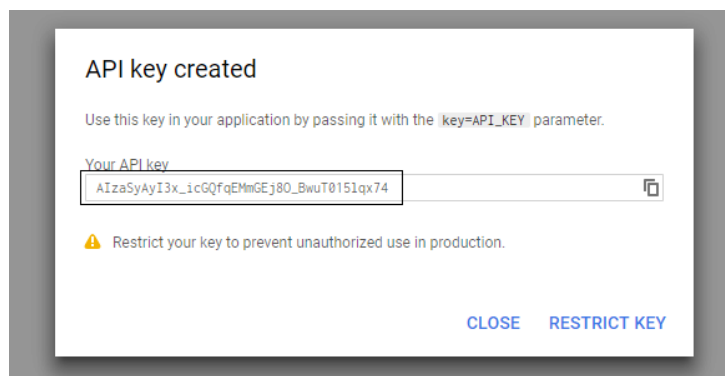
Obrázek 13: Vytvoření API klíče

Zde vybereme 'API key' v nabídce 'Create Credentials'



Obrázek 14: Vytvoření API klíče 2

Toto je náš API klíč

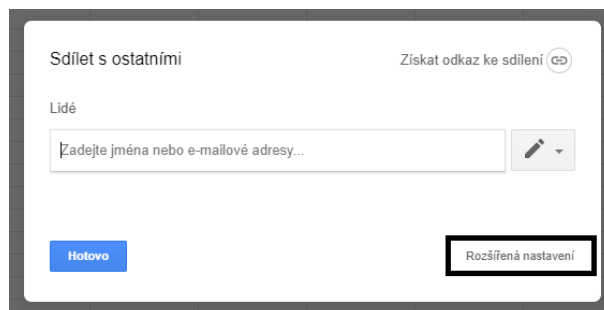


Obrázek 15: Vytvoření API klíče 3

4.2 Nasdílení tabulky

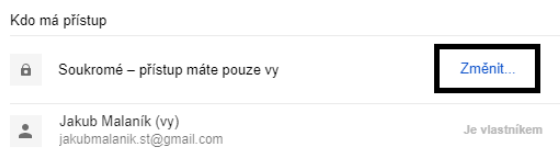
Pro čtení z tabulky pomocí Google Sheets API je ji nutno nasdílet, a to následujícím způsobem:

Tabulku, kterou chceme načíst, otevřeme a v pravo nahoře klikneme na 'Sdílet'. Otevře se nám okno, ve kterém zvolíme 'Rozšířená nastavení'



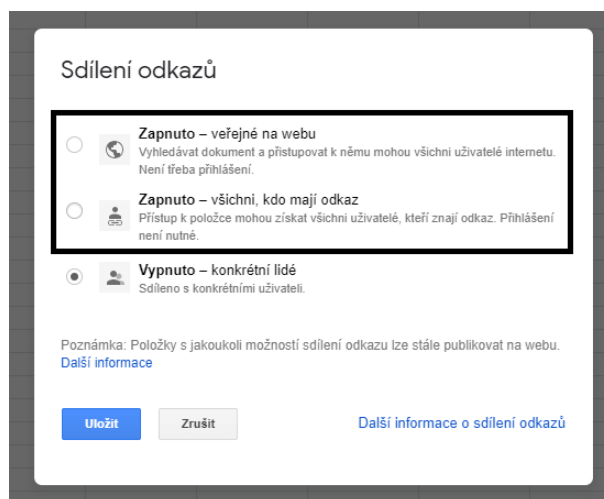
Obrázek 16: Sdílení tabulky

Následně klikneme na 'Změnit'



Obrázek 17: Sdílení tabulky 2

Zde vybereme jednu z následujících dvou možností a uložíme. V tuto chvíli jsme připraveni pro čtení tabulky.



Obrázek 18: Sdílení tabulky 3

4.3 Vyčítání hodnot tabulky

Pro načítání hodnot z tabulky budeme potřebovat Spreadsheet ID, rozsahy A1 notace a API klíč. K načítání hodnot tabulek použijeme program GoogleSheetsValuesToMatlab.m, který se nachází v příloze ve složce GoogleSheetsValuesToMatlab. Pro vytvoření tohoto programu jsem se inspiroval programem, který zmíním na konci práce, viz. [4].

```
1 spreadsheetID='11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws'; %Spreadsheet ID
2 Ranges='List1'; %Anotace A1
3 Key='AIzaSyAyI3x_icGQfqEMmGEj8O_BwuT015lqx74'; %API klíč
```

Zde vidíme parametry, které zadáme před voláním funkce:

GoogleSheetsValuesToMatlab(spreadsheetID,Ranges,Key);

Z těchto hodnot program vygeneruje URL, kterým se volá daná tabulka. Výsledné URL vypadá:

https://sheets.googleapis.com/v4/spreadsheets/1E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws/values/List1?key=AIzaSyAyI3x_icGQfqEMmGEj8O_BwuT015lqx74

Kde červeně zaznačené písmo jsou hodnoty, které zadáváme jako parametry. První parametr představuje Spreadsheet ID viz. 3.1.1.1. Následuje určení rozsahu, viz. 3.1.1.3, a jako poslední API klíč, viz. 4.1. Po spuštění dostaneme tento výstup:

```

1      {
2          "range": "List1!A1:Z1000", %Velikost tabulky
3          "majorDimension": "ROWS", %Výpis hodnot po řádcích
4          "values": [                %Hodnoty
5              [
6                  "zkouška",          %A1
7                  "b1",                %B1
8                  "c1",                %C1
9                  "d1"                 %D1
10             ],
11             [
12                 "a2",                %A2
13                 "b2",                %B2
14                 "c2",                %C2
15                 "D1_"                %D1
16             ]
17         ]
18     }

```

Zde vidíme výpis této tabulky:

	A	B	C	D
1	zkouška	b1	c1	d1
2	a2	b2	c2	D1_
3				
4				
5				
6				

Obrázek 19: Tabulka hodnoty

5 Vyčítání hodnot a vlastností z Google sheets prostřednictvím Matlabu

Čtení hodnot, vlastností hodnot a vlastností tabulky je podobná záležitost, jako čtení samotných hodnot. Rozdíl nastává ve formátu URL a v zadávání parametrů do URL. Nejenže bude potřeba do URL zadat Spreadsheet ID, Anotaci A1 a API klíč, ale i 'Fields', což je parametr, který specifikuje, které vlastnosti budou vypsány. Pokud parametr 'Fields' ne zadáme, budou vypsány vlastnosti tabulky, vlastnosti vybraného listu a sloučené buňky.

Pro načítání hodnot a vlastností z Google Sheets použijeme program `GetGoogleSpreadsheet.m`, který se nachází v příloze ve složce `GoogleSheetsToMatlab`. Pro vytvoření tohoto programu jsem se inspiroval programem, který zmíním na konci práce, viz [4].

```
1 spreadsheetID='11E0PaV01w_e8wopo1cZLIAiDMO6TV3ICvuL8hUeWYws'; %Spreadsheet ID
2 Ranges='List2'; %Anotace A1
3 Fields=''; %Parametr Fields
4 Key='AIzaSyAyI3x_icGQfqEMmGEj8O_BwuT015lqx74'; %API klíč
```

Zde vidíme parametry programu `GetGoogleSpreadsheet.m`, volání funkce:

`GetGoogleSpreadsheet(spreadsheetID ,Ranges, Fields, Key)`

Z těchto hodnot program vygeneruje URL, kterým se volá daná tabulka. Výsledné URL:
https://sheets.googleapis.com/v4/spreadsheets/11E0PaV01w_e8wopo1cZLIAiDMO6TV3ICvuL8hUeWYws?ranges=List1!A1&key=AIzaSyAyI3x_icGQfqEMmGEj8O_BwuT015lqx74

Stejně jako u načítání hodnot, červené písmo značí parametry, které můžeme v programu měnit podle toho, jakou tabulku chceme volat, viz. 4.3.

V tuhle chvíli necháme parametr 'Fields' prázdný pro demonstraci, které vlastnosti lze bez specifikace tohoto parametru z tabulky vyčíst. Po zavolání funkce dostaneme:

```

1
2 { "spreadsheetId": "11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws", %SpeadsheetID
3   "properties": {                                     %Vlastnosti tabulky
4     "title": "Zkouška",                               %Název tabulky
5     "locale": "cs_CZ",                               %Lokace
6     "autoRecalc": "ON_CHANGE",                       %Parametr určující, jakým způsobem
7       budou přepočítávány hodnoty
8     "timeZone": "Europe/Berlin",                     %Časová zóna
9     "defaultFormat": {                               %Defaultní vlastnosti tabulky
10       "backgroundColor": {                          %Barva pozadí buněk
11         "red": 1,"green": 1,"blue": 1               %RGB
12       },
13       "padding": {                                  %Vnitřní mezera mezi okrajem buňky a
14         textem
15       "top": 2,"right": 3,"bottom": 2,"left": 3
16     },
17     "verticalAlignment": "BOTTOM",                  %Vertikální zarovnání
18     "wrapStrategy": "OVERFLOW_CELL",               %Způsob zabalení textu v buňce
19     "textFormat": {                                %Formát textu
20       "foregroundColor": {},                       %Barva textu
21       "fontFamily": "arial,sans,sans-serif",       %Font
22       "fontSize": 10,                             %Velikost písma
23       "bold": false,                              %Tučně
24       "italic": false,                            %Kurzíva
25       "strikethrough": false,                     %Přeškrtnutí
26       "underline": false                          %Podtržení
27     }},
28     "sheets": [{                                    %Vlastnosti listu
29       "properties": {
30         "sheetId": 1489130067,                     %Sheet ID
31         "title": "List2",                          %Název Listu
32         "index": 1,                                %Index
33         "sheetType": "GRID",                       %Typ Listu(defaultní, nelze změnit)
34         "gridProperties": {
35           "rowCount": 1000,                         %Počet řádků v listu
36           "columnCount": 26                         %Počet sloupců v listu
37         }},
38       },
39     ],
40     "spreadsheetUrl": "https://docs.google.com/spreadsheets/d/11
41       E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws/edit" %URL Tabulky
42   }

```

Další možnosti:

- autoRecalc
 - ON_CHANGE . . . přepočítávání po změně
 - Minute . . . přepočítávání po minutě
 - Hour . . . přepočítávání po hodině
- wrapStrategy
 - LEGACY_WRAP . . . Pokud je slovo delší jak řádek, bude oříznuto a další slovo v textu pokračuje na následující řádek. Tato možnost není podporována na všech platformách a je postupně ukončována.
 - CLIP . . . Pokud je text delší jak buňka, bude oříznut. Text nebude zalomen na další řádek, pokud uživatel nevloží ručně nový řádek.
 - WRAP . . . Slovo delší jak buňka je oříznuto a bude pokračovat na dalším řádku.

5.1 Vyčítání určitých vlastností

Vyčítání určitých vlastností tabulky se provádí zadáním parametru 'Fields'. Do tohoto parametru zadáváme, které vlastnosti cheme vyčíst z tabulky. Jsou to:

- spreadsheetId . . . spreadsheet ID
- properties . . . vlastnosti tabulky
- sheets . . . vlastnosti listů
- namedRanges . . . pojmenované rozsahy definované v tabulce
- spreadsheetUrl . . . URL tabulky
- developerMetadata . . . metadata vývojáře spojená s tabulkou

My se dále budeme zabývat 'properties' a 'sheets'.

5.1.1 Properties

Properties		
title	Název tabulky	
locale	Lokace	
autoRecalc	Doba, po kterou se čeká na přepočítávání funkcí	
timeZone	Časová zóna	
defaultFormat	Defaultní formát všech buněk v tabulce	numberFormat, backgroundColor, borders, padding, horizontalAlignment, verticalAlignment, wrapStrategy, textDirection, textFormat, hyperlinkDisplayType, textRotation
iterativeCalculationSetting	Určuje, zda a jak jsou kruhové odkazy vyřešeny iterativním výpočtem	maxIterations, convergenceThreshold

V této tabulce jsou popsány jednotlivé možnosti 'properties'. V prvním sloupci jsou vypsány základní možnosti, ve druhém popis a ve třetím rozvětvení základních možností.

Příklady vypsání určitých vlastností (Parametr 'Fields'):

- Fields=properties.title . . . vypsání názvu tabulky
- Fields=properties.defaultFormat.backgroundColor . . . vypsání defaultní barvy pozadí
- Fields=properties.defaultFormat(borders,textFormat) . . . vypsání defaultního ohraničení buňky a formátu textu

5.1.2 Sheets

Sheets		
properties	Vlastnosti listu	sheetId, title, index, sheetType, gridProperties, hidden, tabColor, rightToLeft
data	Data v buňkách	startRow, startColumn, rowData, rowMetadata, columnMetadata
merges	Sloučené buňky	sheetId, startRowIndex, endRowIndex, startColumnIndex, endColumnIndex
conditionalFormats	Dodatečné formátování	ranges, booleanRule, gradientRule
filterViews	Filter wiews	filterViewId, title, range, namedRangeId, sortSpecs, criteria
protectedRanges	Chráněné rozsahy	maxIterations, convergenceThreshold
basicFilter	Filtr listu	range, sortSpecs, criteria
charts	Grafy	chartId, spec, position
bandedRanges	Pásmové rozsahy	bandedRangeId, range, rowProperties, columnProperties
developerMetadata	Metadata vývojáře	metadataId, metadataKey, metadataValue, location, visibility
rowGroups	Skupiny řádků	range, depth, collapsed
columnGroups	Skupiny sloupců	range, depth, collapsed

Tato tabulka popisuje možnosti 'Sheets'. 'Sheets' slouží k vypisování vlastností jednotlivých listů tabulky. Příklady vypsání určitých vlastností (Parametr 'Fields'):

- Fields=sheets . . . vypsání všeho, co je v 'sheets'
- Fields=sheets.properties . . . vypsání vlastností listu
- Fields=sheets.charts(chartId,spec) . . . vypsání ID a vlastností grafu

V následující tabulce je popsána možnost vypsání dat z buněk, jako je barva textu, barva pozadí, ohraničení buněk apod.

rowData.values		
userEnteredValue	Hodnota vložená uživatelem	numberValue, stringValue, boolValue, formulaValue, errorValue
effectiveValue	Data v buňkách	numberValue, stringValue, boolValue, formulaValue, errorValue
formattedValue	Formátovaná hodnota	
userEnteredFormat	Formátování zadané uživatelem	numberFormat, backgroundColor, borders, padding, horizontalAlignment, verticalAlignment, wrapStrategy, textDirection, textFormat, hyperlinkDisplayType, textRotation
effectiveFormat	Formátování využívané buňkou	numberFormat, backgroundColor, borders, padding, horizontalAlignment, verticalAlignment, wrapStrategy, textDirection, textFormat, hyperlinkDisplayType, textRotation
hyperlink	Hypertextový odkaz	
note	Poznámka	
textFormatRuns		startIndex, format
dataValidation	Validace dat buňky	condition, inputMessage, strict, showCustomUi
pivotTable	Pivot tabulka ukotvena k buňce	source, rows, columns, criteria, values, valueLayout

Příklady vypsání určitých vlastností (Parametr 'Fields'):

- Fields=sheets.data.rowData.values . . . vypsání všeho, co je v 'values'
- Fields=sheets.data.rowData.values.effectiveFormat . . . vypsání formátování buněk
- Fields=sheets.data.rowData.values(effectiveValue,note) . . . vypsání efektivní hodnoty a poznámky k buňkám

5.1.3 Výpis užitečných vlastností

Již vytvořenou tabulku, jejíž Spreadsheets ID a API klíč jsou vypsány výše viz. 5. Místo 'Fields' a 'Ranges' zadáme:.

```
1 Ranges='List2!A1:C1';  
2 Fields='sheets.data.rowData.values(formattedValue,userEnteredFormat),sheets.  
   data.rowData.values.effectiveFormat.textRotation.angle,sheets.merges,sheets  
   .data(rowMetadata,columnMetadata)';
```

Znamená to, že budeme vyžadovat formátovanou hodnotu buňky(string), formátování zadávané uživatelem, úhel pod kterým je text otočen, informace o tom, které buňky jsou sloučeny a metadata řádků a sloupců, což vypíše velikosti buněk v pixelech. Tyto informace budou brány z prvních tří buněk řádku jedna.

Tabulku jsem modifikoval. V buňce A1 jsem provedl co nejvíce změn, jako je ohrazení buňky, změna barvy pozadí, změna fontu apod. Buňky B1 a C1 byly sloučeny. Zde je tabulka, jejíž vlastnosti a hodnoty budeme vyžadovat:

	A	B	C
1	1000 2025		

Obrázek 20: Tabulka vyčítání vlastností

Zavoláme funkci `GetGoogleSpreadsheet(spreadsheetID,Ranges,Fields,Key)` se změněnými parametry 'Fields' a 'Ranges':

```

1  {"sheets": [{
2  "data": [{
3    "rowData": [{
4      "values": [{
5        "formattedValue": "VšB 2325",           %String hodnota buňky
6        "userEnteredFormat": {                 %Formátování zadané uživatelem
7          "numberFormat": {                     %Formát čísel
8            "type": "NUMBER",                   %Typ čísla
9            "pattern": "#,##0.00;(##0.00)",      %Vzorec formátu
10         "backgroundColor": {"red": 0.8},       %Barva pozadí
11         "borders": {                           %Ohraničení
12           "top": {                             %Vrchní ohraničení
13             "style": "SOLID_THICK",            %Plná čára
14             "width": 3,                        %Tloušťka čáry
15             "color": {"blue": 1}},             %Barva čáry
16           "bottom": {                          %Spodní ohraničení
17             "style": "SOLID_THICK",
18             "width": 3,
19             "color": {"blue": 1}},
20           "left": {                            %Levé ohraničení
21             "style": "SOLID_THICK",
22             "width": 3,
23             "color": {"blue": 1}},
24           "right": {                          %Pravé ohraničení
25             "style": "SOLID_THICK",
26             "width": 3,
27             "color": {"blue": 1}},
28           "horizontalAlignment": "CENTER",      %Horizontální zarovnání
29           "verticalAlignment": "MIDDLE",        %Vertikální zarovnání
30           "wrapStrategy": "WRAP",              %Zabalení textu v buňce
31           "textFormat": {                      %Formát textu
32             "foregroundColor":
33               {"red": 1,"green": 1},           %Barva textu
34             "fontFamily": "Lobster",           %Font
35             "fontSize": 18,                    %Velikost písma
36             "bold": true,                      %Tučné písmo
37             "italic": true},                  %Kurzíva
38             "textRotation": {                  %Rotace textu
39               "vertical": false}}},
40         "effectiveFormat": {                   %Efektivní formát
41           "textRotation": {                    %Rotace textu
42             "angle": -90 }}}},                %Úhel rotace
43       "rowMetadata": [{                       %Velikost řádků v pixelech
44         "pixelSize": 121}],
45       "columnMetadata": [{                   %Velikost sloupců v pixelech
46         "pixelSize": 155},{
47         "pixelSize": 166},{
48         "pixelSize": 100}}]},
49     "merges": [{                             %Sloučené buňky
50       "sheetId": 1489130067,                 %sheet ID
51       "startRowIndex": 0,"endRowIndex": 1,"startColumnIndex": 1,"endColumnIndex": 3}}]}
52                                           %Rozsah sloučených buněk

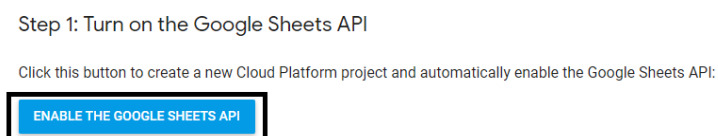
```

6 Nahrávání dat do Google Sheets prostřednictvím Matlabu

Jako při načítání hodnot z Google Sheets do Matlabu budeme potřebovat pro nahrávání hodnot spreadsheet ID a sheet ID. Navíc si budeme muset vytvořit Client ID a Client Secret, což jsou hodnoty, které později využijeme pro vygenerování přístupových tokenů. Tyto tokeny nám pak umožní export dat z Matlabu do Google sheets.

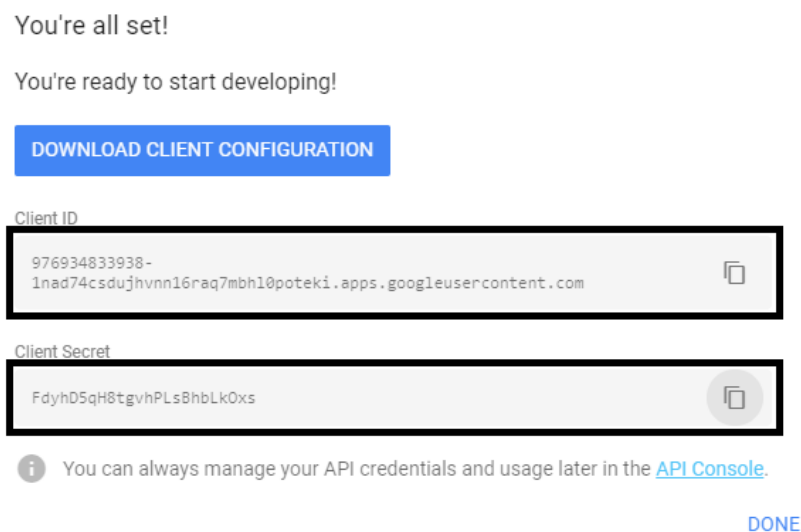
6.1 Vytvoření Client ID a Client Secret

Jako první se přihlásíme v prohlížeči na svůj Google account. Poté navštívíme tuto adresu: <https://developers.google.com/sheets/api/quickstart/java>. Klikneme na 'ENABLE THE GOOGLE SHEETS API'



Obrázek 21: Vygenerování Client Secret a Client ID

Po chvíli se otevře okno, kde je Client ID a Client Secret vygenerováno.



Obrázek 22: Vygenerování Client Secret a Client ID 2

6.2 Vygenerování přístupových tokenů

Pro vygenerování tokenů použijeme program RunOnce.m v příloze ve složce MatlabToSheets. Pro vytvoření programu jsem se inspiroval kódem, který uvedu na konci práce, viz. [5].

```
1 client_secret='FdyhD5qH8tgvhPLsBhbLk0xs'; %Client Secret
2 client_id='976934833938-1nad74csdujhvnn16raq7mbhl0poteki.apps.googleusercontent
    .com'; %Client ID
```

Zde vidíme parametry funkce: RunOnce(client_id, client_secret)

Tedy Client ID a Client Secret, které jsme si vytvořili výše viz. 6.1. Po zavolání funkce bude do command window vypsáno:

```
1 Otevřete prohlížeč a pokračujte na: https://www.google.com/device
2 Zadejte kód: NMYW-NPTR
3 Stiskněte jakoukoliv klávesu, pokud jste úspěšně dokončili předchozí kroky
```

To znamená, že v prohlížeči na adrese <https://www.google.com/device> zadáme kód 'NMYW-NPTR' a následně povolíme přístup čtení, úpravy, vytváření a mazání vašich tabulek na disku Google.

Poté v command window stiskneme jakoukoliv klávesu a ve složce MatlabToSheets bude vygenerován soubor google_tokens.mat, ve kterém jsou uloženy přístupové tokeny.

6.3 Nahrávání hodnot

Po vygenerování přístupových tokenů můžeme přejít v příloze ve složce MatlabToSheets na program mat2sheets.m. Pro zapisování dat musíme mít soubor google_tokens.mat ve složce s programem mat2sheets.m. Pro vytvoření tohoto programu jsem se inspiroval již vytvořeným programem, který zmíním na konci práce viz. [5].

Pro nahrávání hodnot budeme volat funkci:
mat2sheets(spreadsheetID, sheetID, pos, d), kdy:

- spreadsheetID . . . spreadsheet ID(String)
- sheetName . . . jméno tabulky(String)
- pos . . . pole[sloupec řádek] pro indikaci, kam začít zapisovat data. Indexování začíná od jedné.
- d . . . pole dat [] pro zápis do tabulky
 - Zápis int . . . [1 2 3]
 - Zápis double . . . [1.23 2.23 3.23]
 - Zápis String(1 znak) . . . [{'a'} {'b'} {'c'}]
 - Zápis String(více znaků) . . . [{'abc'} {'def'} {'ghi'}]

6.3.1 Praktická ukázka

Pro tuto ukázku budeme používat následující hodnoty, které si nahrajeme do workspace:

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';  
2 sheetName = 'ListHodnoty';
```

Jako ukázku nahraji pole o rozměru 4x10, kde ukážu jakým způsobem nahrávat hodnoty. Voláme funkci:

```
1 mat2sheets(spreadsheetID, sheetName, ... %Specifikace tabulky  
2 [2 2], ...%Určení počátku zápisu  
3 [{'INT'} {'DOUBLE'} {'SYMBOL'} {'STRING'}]; ...%Záhlaví tabulky  
4 1 1.1 {'a'} {'ab'};2 2.22 {'b'} {'cd'};3 3.333 {'c'} {'ef'}; ...%2.-4. řádek  
5 5 5.5555 {'d'} {'gh'};7 7.7777 {'e'} {'ch'};11 11.1111 {'f'} {'ij'}; ...%5.-7.  
6 13 13.1313 {'g'} {'kl'};17 17.1717 {'h'} {'mn'};19 19.1919 {'i'} {'op'}}) %8.-10.
```

V prvních dvou parametrech definujeme tabulku a list, na který chceme zapisovat hodnoty(spreadsheetID, sheetName). Třetí parametr je indikace, kde začít zapisovat([2,2]) a poslední parametr je pole hodnot, které budeme importovat do tabulky(řádek '3' až '6'). Výsledná tabulka vypadá následovně:

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 23: Tabulka zápis hodnot

6.4 Změna barvy pozadí

Pro změnu barvy pozadí a další editace budeme používat program `matProperties2sheets.m`, který se nachází v příloze ve složce `MatlabPropertiesToSheets` a vychází z programu `mat2sheets`, viz. 6.3.

Ještě před spuštěním je potřeba vygenerovat přístupové tokeny, viz 6.2. Tentokrát se program `RunOnce.m` nachází ve složce s `matProperties2sheets.m`. Podobně jako u nahrávání hodnot, budeme volat funkci:

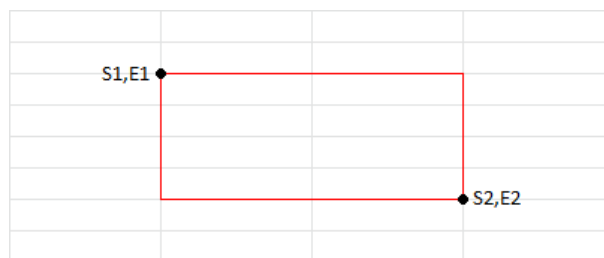
`matProperties2sheets(spreadsheetID, sheetID, D, p)`

- `spreadsheetID` . . . spreadsheet ID(string)
- `sheetID` . . . sheetID(string)
- `D` . . . specifikace úlohy jakou funkce bude vykonávat (hodnoty jsou citlivé na velikost písmen)
 - `BarvaPozadi` . . . změna barvy pozadí
 - `BarvaTextu` . . . změna barvy textu
 - `FormatTextu` . . . změna formátu textu
 - `Ohranici` . . . editace ohraničení
 - `Grafy` . . . grafy
- `p` . . . Pole parametrů určující vlastnoti dané úlohy

6.4.1 Pole 'p' pro změnu barvy pozadí

Pole 'p' je strukturováno následovně: [`S1 E1 S2 E2 C1 C2 C3`]

- `S1,E1,S2,E2` . . . Souřadnice oblasti pro změnu barvy pozadí. Indexování začíná od jedné.



Obrázek 24: Tabulka souřadnice

- `RGB1,RGB2,RGB3` . . . Barva v RGB formátu

6.4.2 Praktická ukázka

Tabulku pro ukázku použiju tu, již jsem si vytvořil při nahrávání hodnot viz. 6.3.1.

```
1 spreadsheetID ='11EOPaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID ='370178304';
3 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
4 {'BarvaPozadi'}, ...%Určení úlohy pro vykonání
5 [2 2 6 3 ...%Souřadnice oblasti[S1,S2,E1,E2]
6 255 255 125 ])%Žlutá barva ve formátu RGB [RGB1 RGB2 RGB3]
```

Vysledek vypadá:

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 25: Tabulka změna barvy pozadí

6.5 Změna barvy textu

Parametry pro volání funkce pro změnu barvy textu jsou téměř totožné jako při editaci barvy pozadí. Změna nastává jen v parametru 'D', který bude 'BarvaTextu' viz 6.4. Parametry pro pole 'p' zůstávají stejné, to znamená [S1 E1 S2 E2 C1 C2 C3], kde S1,E1,S2,E2 jsou souřadnice a RGB1,RGB2,RGB3 je barva viz 6.4.1.

6.5.1 Praktická ukázka

V ukázce budeme pokračovat s tabulkou, kterou máme vytvořenou viz. 6.4.2.

```
1 spreadsheetID ='11EOPaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID ='370178304';
3 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
4 {'BarvaTextu'}, ...%Určení úlohy pro vykonání
5 [2 2 6 3 ...%Souřadnice oblasti[S1,S2,E1,E2]
6 250 0 0 ])%Červená barva ve formátu RGB [RGB1 RGB2 RGB3]
```

Výsledek:

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 26: Tabulka změna barvy textu

6.6 Změna formátu textu

Změna formátu textu znamená editaci:

- velikosti písma
- fontu
- tučného písma
- kurzívy písma
- přeškrtnutí textu
- podtržení textu

Pro editaci formátu voláme funkci: `matProperties2sheets(spreadsheetID, sheetID, D ,p)` viz. 6.4, to znamená parametr 'D' bude 'FormatTextu' a pole 'p':

6.6.1 Pole 'p' pro změnu formátu textu

Pole 'p' je strukturováno následovně [S1 E1 S2 E2 Size Font Bold Ital Strike Under]

- S1,E1,S2,E2 . . . souřadnice oblasti pro změnu formátu viz 6.4.1
- Size . . . velikosti písma
- Font . . . font, například {'Arial'} (hodnoty jsou citlivé na velikost písmen)
- Bold, Ital, Strike, Under . . . tučné písmo, kurzíva písma, přeškrtnutí textu, podtržení textu. Tyto parametry nabývají hodnoty 1 (true) nebo 0 (false).

6.6.2 Praktická ukázka

Tak jako ve všech ostatních ukázkách, budeme pokračovat v již vytvořené a editované tabulce viz. 6.5.1.

```
1 spreadsheetID ='11EOPaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID ='370178304';
```

Budeme volat dvě funkce. První:

```
1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'FormatTextu'}, ...%Určení úlohy pro vykonání
3 [2 2 6 3 ...%Souřadnice oblasti(Záhlaví tabulky) [S1,S2,E1,E2]
4 15 ...%Velikost textu
5 {'Courier new'} ...%Font textu
6 1 0 0 1 ])%[Tučně(True) Kurzíva(False) Přeškrtnutí(False) Podtržení(True)]
```

Tato funkce se zaměřuje na záhlaví tabulky, kde bude změněna velikost písma a font písma. Poté bude text změněn na tučný a podtržený. Po zavolání funkce bude tabulka vypadat následovně:

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 27: Tabulka změna formátu 1

Druhá funkce:

```

1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'FormatTextu'}, ...%Určení úlohy pro vykonání
3 [2 3 6 12 ...%Souřadnice oblasti(Hodnoty Tabulky) [S1,S2,E1,E2]
4 11 ...%Velikost textu
5 {'Georgia'} ...%Font textu
6 0 1 0 0 ])%[Tučně(False) Kurzíva(True) Přeškrtnutí(False) Podtržení(False)]

```

Druhá funkce řeší zbytek tabulky(hodnoty). Bude změněn font, velikost textu a text je psán kurzívou.

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		<i>1</i>	<i>1,1</i>	<i>a</i>	<i>ab</i>	
4		<i>2</i>	<i>2,22</i>	<i>b</i>	<i>cd</i>	
5		<i>3</i>	<i>3,333</i>	<i>c</i>	<i>ef</i>	
6		<i>5</i>	<i>5,5555</i>	<i>d</i>	<i>gh</i>	
7		<i>7</i>	<i>7,7777</i>	<i>e</i>	<i>ch</i>	
8		<i>11</i>	<i>11,1111</i>	<i>f</i>	<i>ij</i>	
9		<i>13</i>	<i>13,1313</i>	<i>g</i>	<i>kl</i>	
10		<i>17</i>	<i>17,1717</i>	<i>h</i>	<i>mn</i>	
11		<i>19</i>	<i>19,1919</i>	<i>i</i>	<i>op</i>	
12						

Obrázek 28: Tabulka formátu 2

6.7 Editace ohraničení

Jako u předchozích bodů, voláme funkci: `matProperties2sheets(spreadsheetID, sheetID ,D ,p)`.
'D' je 'Ohranici' a 'p':

6.7.1 Pole 'p' pro editaci ohraničení

Pole 'p' je strukturováno následovně: [S1 E1 S2 E2 Position Style Width RGB1 RGB2 RGB3]

- S1,E1,S2,E2 . . . souřadnice oblasti pro změnu formátu viz 6.4.1
- Position . . . pozice ohraničení (hodnoty jsou citlivé na velikost písmen)
 - top . . . horní ohraničení
 - bottom . . . spodní ohraničení
 - left . . . levé ohraničení
 - right . . . pravé ohraničení
 - innerHorizontal . . . vnitřní horizontální ohraničení
 - innerVertical . . . vnitřní vertikální ohraničení
- Style . . . Styl čáry (Hodnoty jsou citlivé na velikost písmen)
 - DOTTED . . . tečkovaná čára hustá(.....)
 - DASHED . . . tečkovaná čára(. . .)
 - SOLID . . . plná čára
 - SOLID_MEDIUM . . . plná čára(tenká)
 - SOLID_THICK . . . plná čára(tlustá)
 - DOUBLE . . . dvojitá čára
- Width . . . tloušťka čáry (hodnota 1-3, ale jen u stylu čáry 'SOLID', u ostatních stylů tato hodnota změnit nejde, zůstává '1')
- RGB1,RGB2,RGB3 . . . barva čáry v RGB formátu

6.7.2 Praktická ukázka

Budeme pokračovat s vytvořenou a editovanou tabulkou z předchozích bodů viz. 6.6.2.

```
1 spreadsheetID ='11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID ='370178304';
```

Tentokrát budeme volat 7 funkcí:

- Zde je řešeno ohrazení celé tabulky dokola tlustou modrou čarou.

```
1 matProperties2sheets(sheetID, sheetID, ...%Specifikace tabulky
2 {'Ohranici'}, ...%Určení úlohy pro vykonání
3 [2 2 6 12 ...%Souřadnice oblasti(Celá tabulka)[S1,S2,E1,E2]
4 {'top'} ...%Pozice ohraničení(vrchní)
5 {'SOLID'} ...%Styl čáry(plná čára)
6 3 ...%Tloušťka čáry
7 0 0 255 ])%Modrá barva čáry v RGB formátu [RGB1 RGB2 RGB3]
8
9 matProperties2sheets(sheetID, sheetID,{'Ohranici'},[2 2 6 12 {'right'}
   {'SOLID'} 3 0 0 255])
10 matProperties2sheets(sheetID, sheetID,{'Ohranici'},[2 2 6 12 {'bottom'}
   {'SOLID'} 3 0 0 255])
11 matProperties2sheets(sheetID, sheetID,{'Ohranici'},[2 2 6 12 {'left'} {
   'SOLID'} 3 0 0 255])
```

Jediný parametr, který u funkce měníme, je pozice ohraničení, což značí, z které strany bude tabulka ohraničena. Parametry 'top', 'right', 'bottom' a 'left' ohraničíme celou tabulku.

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 29: Tabulka ohraničení 1

- Tato funkce pracuje se záhlavím tabulky, kde pozice ohraničení je nastavena na 'bottom'. To znamená, že bude ohraničena spodní část záhlaví čarou barvy mordé.

```

1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'Ohranicieni'}, ...%Určení úlohy pro vykonání
3 [2 2 6 3 ...%Souřadnice oblasti(záhlaví)[S1,S2,E1,E2]
4 {'bottom'} ...%Pozice ohraničení(spodní)
5 {'SOLID'} ...%Styl čáry(plná čára)
6 3 ...%Tloušťka čáry
7 0 0 255 ])%Modrá barva čáry v RGB formátu [RGB1 RGB2 RGB3]

```

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 30: Tabulka ohraničení 2

- Poslední 2 funkce se zaměřují na zbytek naší tabulky. Parametry 'innerHorizontal' a 'innerVertical' značí horizontální a vertikální ohraničení uvnitř tabulky. 'DASHED' a 'DOTTED' jsou parametry pro styl čar.

```

1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'Ohranicieni'}, ...%Určení úlohy pro vykonání
3 [2 3 6 12 ...%Souřadnice oblasti(záhlaví)[S1,S2,E1,E2]
4 {'innerHorizontal'} ...%Pozice ohraničení(vnitřní horizontální)
5 {'DASHED'} ...%Styl čáry(tečkovaná)
6 1 ...%Tloušťka čáry
7 255 100 0 ])%Oranžová barva čáry v RGB formátu [RGB1 RGB2 RGB3]

```

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 31: Tabulka ohraničení 3

```

1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'Ohranicieni'}, ...%Určení úlohy pro vykonání
3 [2 3 6 12 ...%Souřadnice oblasti(záhlaví)[S1,S2,E1,E2]
4 {'innerVertical'} ...%Pozice ohraničení(vnitřní horizontální)
5 {'DOTTED'} ...%Styl čáry(tečkovaná hustá)
6 1 ...%Tloušťka čáry
7 0 255 255 ])%Oranžová barva čáry v RGB formátu [RGB1 RGB2 RGB3]

```

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1,1	a	ab	
4		2	2,22	b	cd	
5		3	3,333	c	ef	
6		5	5,5555	d	gh	
7		7	7,7777	e	ch	
8		11	11,1111	f	ij	
9		13	13,1313	g	kl	
10		17	17,1717	h	mn	
11		19	19,1919	i	op	
12						

Obrázek 32: Tabulka ohraničení 4

6.8 Přidávání grafů

Jako všechny předchozí operace, funkce pro vytvoření grafu voláme:

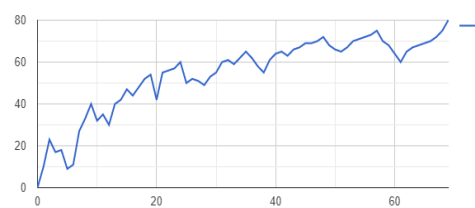
`matProperties2sheets(spreadsheetID, sheetID, D, p)`. 'D' je nastaveno na 'Grafy' a pole 'p':

6.8.1 Pole 'p' pro editaci ohraničení

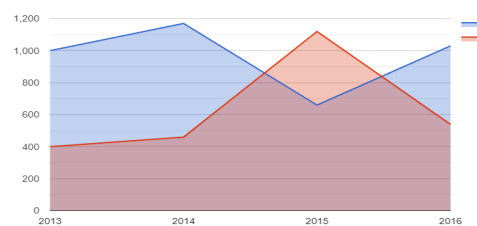
Pole 'p' je strukturováno následovně: [S1 E1 S2 E2 Title CharType LegendPosition TitleBottom TitleLeft ChartPositionX ChartPositionY Width Height]

- S1,E1,S2,E2 . . . souřadnice oblasti dat pro vytvoření grafu viz 6.4.1
- Title . . . jméno grafu
- CharType . . . typ grafu (hodnoty jsou citlivé na velikost písmen)

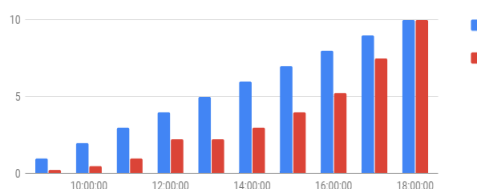
– LINE(čárový graf)



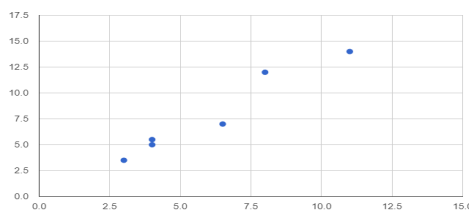
– AREA(plošný graf)



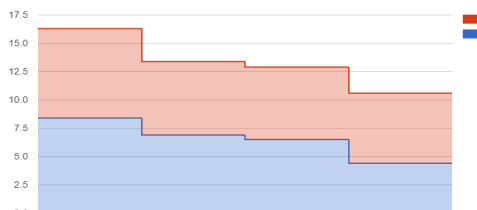
– COLUMN(sloupcový graf)



– SCATTER(bodový graf)



– STEPPED_AREA(schodový graf)



- LegendPosition . . . pozice legendy (hodnoty jsou citlivé na velikost písmen)
 - BOTTOM_LEGEND . . . legenda je zobrazena ve spodní části grafu
 - LEFT_LEGEND . . . legenda je zobrazena v levé části grafu
 - RIGHT_LEGEND . . . legenda je zobrazena v pravé části grafu
 - TOP_LEGEND . . . legenda je zobrazena ve vrchní části grafu
 - NO_LEGEND . . . bez legendy
- TitleBottom . . . jméno spodní osy
- TitleBottom . . . jméno levé osy
- ChartPositionX,ChartPositionY . . . počáteční souřadnice, na něž bude přichycen levý vrchní roh grafu (sloupec * řádek)
- Width,Height . . . šířka a výška grafu (v pixelech)

6.8.2 Praktická ukázka

Pro ukázkou použijí 2 různé typy grafů, které budou brát výchozí hodnoty ze stejné tabulky. Je to tabulka, kterou jsme již používali v předchzích úlohách, jejíž hodnoty byly pozměněny:

	A	B	C	D	E	F
1						
2		SYMBOL	DOUBLE	INT	INT2	
3		<i>a</i>	1,1	1	1	
4		<i>b</i>	2,22	2	8	
5		<i>c</i>	3,333	3	10	
6		<i>d</i>	5,5555	5	15	
7		<i>e</i>	7,7777	7	20	
8		<i>f</i>	11,1111	11	30	
9		<i>g</i>	13,1313	13	40	
10		<i>h</i>	17,1717	17	55	
11		<i>i</i>	19,1919	19	60	
12						

Obrázek 33: Tabulka změna hodnot

První a třetí sloupce byly prohozeny a čtvrtý sloupec byl vyměněn za numerické hodnoty.

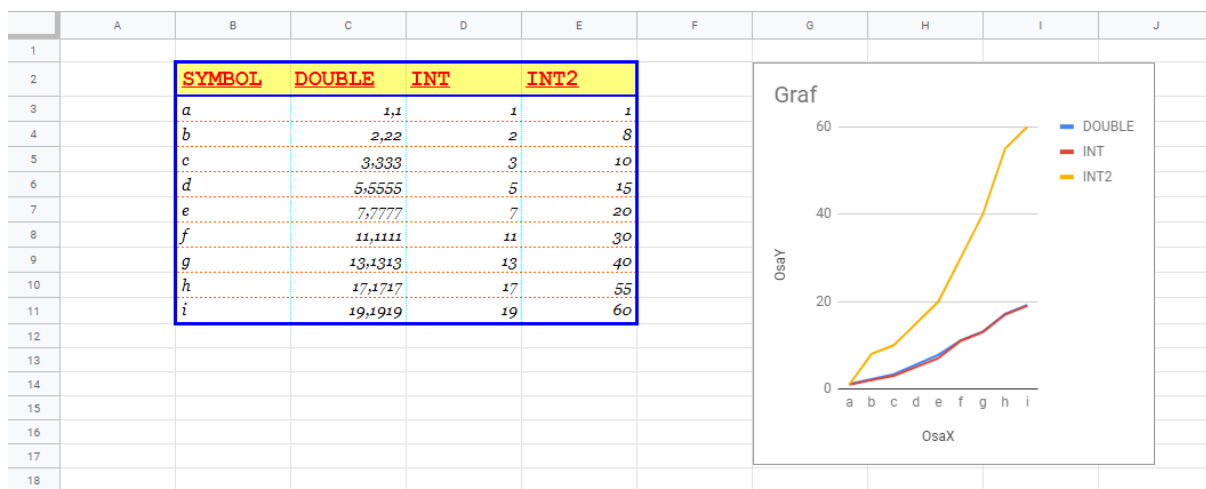
Při zadávání souřadnic oblastí, ze kterých jsou brána data pro vytvoření tabulky, musíme vědět, že první sloupec tabulky bude brán jako doména (hodnoty 'a' až 'i'). Zbylé sloupce budou brány jako data vyobrazená v grafu, jejichž hodnoty v záhlaví (Stringy) budou vypsány jako legendy.

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID = '370178304';
```

Volání první funkce bude vypadat:

```
1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'Grafy'}, ...%Určení úlohy pro vykonání
3 [2 2 6 12 ...%Souřadnice oblasti(výchozí data pro vytvoření grafu) [S1,S2,E1,E2]
4 {'GrafJedna'} ...%Jméno grafu
5 {'LINE'} ...%Typ grafu
6 {'Right_LEGEND'} ...%Pozice legendy
7 {'OsaX'} ...%Jméno spodní osy
8 {'OsaY'} ...%Jméno levé osy
9 2 7 ...%Souřadnice, na kterých bude vytvořen graf(sloupec * řádek)
10 350 350])%Šířka * výška(velikost tabulky v pixelech)
```

Výsledný graf bude vypadat:



Obrázek 34: Tabulka grafy 1

V levém horním rohu vidíme název grafu a na pravé straně je vypsána legenda. Na levé a spodní straně jsou názvy os a data z druhého až čtvrtého sloupce jsou vyobrazena v grafu.

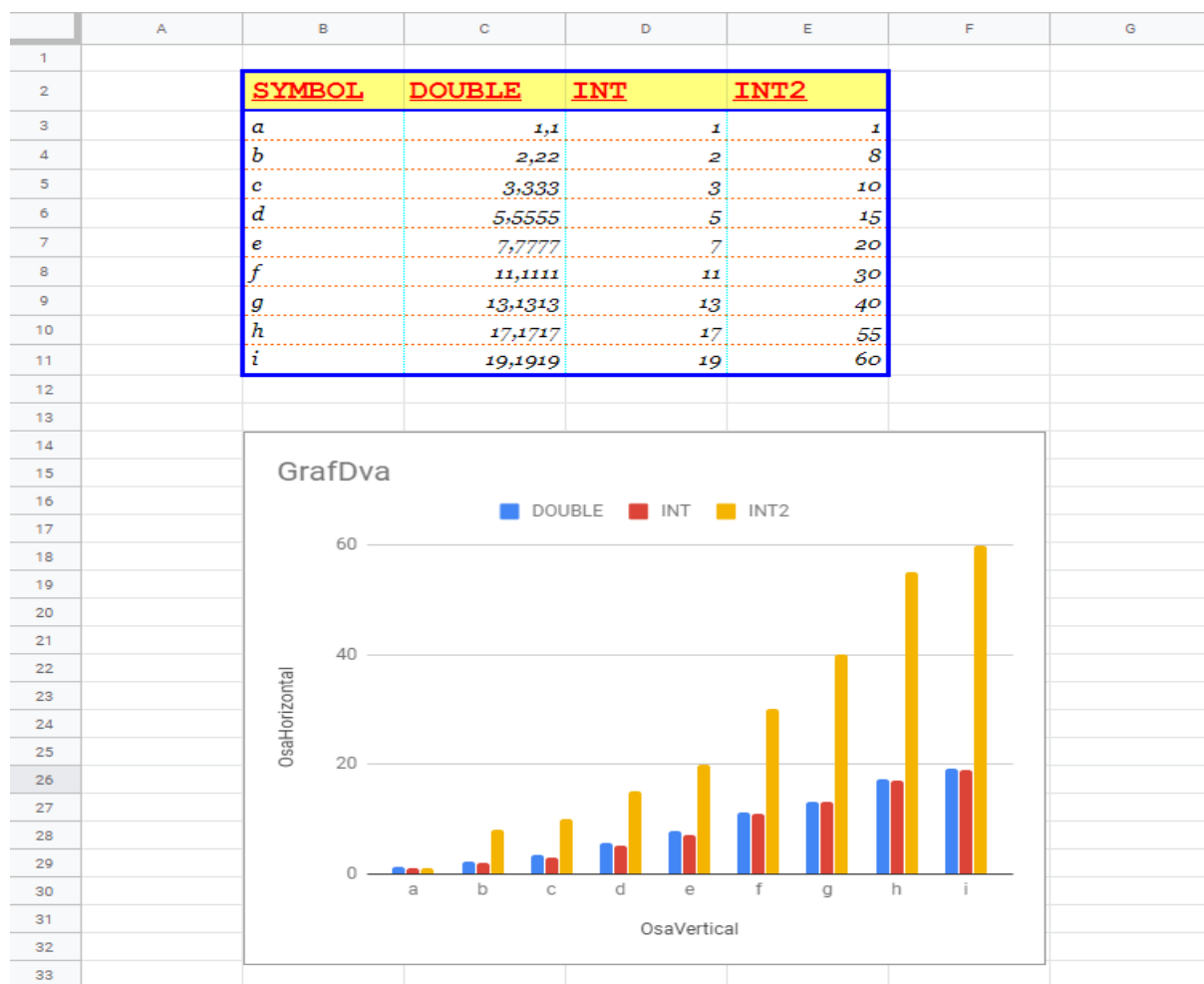
Druhý graf:

```

1 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
2 {'Grafy'}, ...%Určení úlohy pro vykonání
3 [2 2 6 12 ...%Souřadnice oblasti(výchozí data pro vytvoření grafu)[S1,S2,E1,E2]
4 {'GrafDva'} ...%Určení úlohy pro vykonání
5 {'COLUMN'} ...%Typ grafu
6 {'TOP_LEGEND'} ...%Pozice legendy
7 {'OsaVertical'} ...%Jméno spodní osy
8 {'OsaHorizontal'} ...%Jméno levé osy
9 14 2 ...%Souřadnice, na kterých bude vytvořen graf
10 500 400])%Šířka * výška(velikost tabulky)

```

Oproti předchozímu grafu jsem změnil všechny vlastnosti grafu jako jsou: název grafu, typ grafu, pozice legendy, jména os, pozici pro vytvoření grafu a velikost grafu. Výsledek:



Obrázek 35: Tabulka grafy 2

7 Operace s listy Google Sheets prostřednictvím Matlabu

Pro operaci s listy Google Sheets budeme využívat program `matProperties2sheets.m`, který jsme využívali i pro editaci tabulky viz 6.4.

Volání funkce bude také stejné jako u předchozích úloh viz 6.4:

`matProperties2sheets(spreadsheetID, sheetID, D, p)`. Parametr `D`:

- `PridatList` . . . přidání listu
- `SmazatList` . . . smazání listu
- `PrejmenovatList` . . . přejmenování listu
- `KopirovatList` . . . zkopírování listu

Pole `'p'` zadáváme jen v případě, pokud přidáváme list nebo přejmenováváme list. V tomto případě zadáváme jen nové jméno listu. Jinak parametr `'p'` zůstává prázdný.

7.1 Praktické ukázky

Zde si ukážeme, jak jednotlivé úlohy volat.

7.1.1 Přidání listu

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2
3 matProperties2sheets(spreadsheetID, [], ...%Specifikace tabulky. Sheed ID bude
   prázdné pole, jelikož ho u vytváření listu nepotřebujeme
4 {'PridatList'}, ...%Určení úlohy pro vykonání
5 {'ListJedna'}) %Jméno nového listu
```

Přidaný list bude umístěn vpravo vedle posledního listu. Pokud se pokusíme přidat list se jménem, které již existuje, list přidán nebude.

7.1.2 Přejmenování listu

V předchozím bodě jsem vytvořil list s Sheed ID: `'1201898094'`. Nyní tento list přejmenujeme:

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID = '1201898094';
3
4 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
5 {'PrejmenovatList'}, ...%Určení úlohy pro vykonání
6 {'ListNovy'}) %Jméno nového listu
```

Přejmenování názvu listu na jméno již použité není možné, akce provedena nebude.

7.1.3 Zkopírování listu

Při kopírování listu zadáváme Sheet ID listu, který chceme zkopírovat. Po vykonání funkce bude vytvořen nový list se všemi hodnotami a vlastnostmi kopírovaného listu. Před název zkopírovaného listu bude přidáno 'Kopie listu'. Pokud kopírování provedeme znovu na stejný list, akce bude provedena a za název kopie bude přidáno pořadové číslo začínající od '1'. Tedy 'Kopie listu NázevListu 1'.

Zkopírujeme tedy list již vytvořený z předchozího bodu, jehož název bude 'Kopie listu ListNovy':

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID = '1201898094';
3
4 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
5 {'KopirovatList'}) %Určení úlohy pro vykonání
```

7.1.4 Smazání listu

Nyní tento zkopírovaný list, který má Sheet ID: '991472143' smažeme:

```
1 spreadsheetID = '11E0PaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID = '991472143';
3
4 matProperties2sheets(spreadsheetID, sheetID, ...%Specifikace tabulky
5 {'SmazatList'}) %Určení úlohy pro vykonání
```

8 Práce s Google Sheets prostřednictvím Scilabu

Práce s Google Sheets prostřednictvím Scilabu je hodně podobná práci s Matlabem. V této sekci budu popisovat rozdíly, které přináší používání Scilabu. Pokud nezmíním nějaký krok, znamená to, že postup je stejný ve Scilabu jako v Matlabu.

8.1 Vyčítání hodnot

Pro tento účel budeme používat program `GoogleSheetsValuesToScilab.sci` v příloze ve složce Scilab. Vycházím z programu, který jsem využíval v kapitole 4.3.

Ještě než začneme pracovat s programy ve Scilabu, je nutné nastavit parametr `'Folder'`, což je absolutní cesta ke složce `'Scilab'` v našem počítači. V mém případě: `Folder='E:\BP\Scilab'`; Tuto absolutní cestu budou všechny programy využívat.

V našem případě voláme funkci:

```
1 GoogleSheetsValuesToScilab(spreadsheetID,Ranges,Key)
```

Pro ukázkou použijeme parametry a tabulku jako v kapitole 4.3. Výsledek bude vypsán do konzole i do souboru `GoogleSheetsValuesResult.txt`, který se nachází ve složce s programem. Výsledek je vypsán do souboru, jelikož v některých případech je výstup vypsán do konzole špatně.

```
1      {
2      "range": "List1!A1:Z1000",
3      "majorDimension": "ROWS",
4      "values": [
5      [
6          "zkouška",
7          "b1",
8          "c1",
9          "d1"
10     ],
11     [
12         "a2",
13         "b2",
14         "c2",
15         "D1_"
16     ]
17     ]
18 }
```

Zde vidíme výstup po zavolání funkce. Shoduje se s výsledkem z Matlabu, viz 4.3

8.2 Vyčítání hodnot a vlasností

Pro vyčítání hodnot a vlasností využijeme program `GetGoogleSpreadsheet.sci`, který se nachází v příloze ve složce `Scilab`. Program je stejný jako v Matlabu viz 5, akorát modifikovaný pro Scilab. Pro vyčítání použijeme stejný postup jako v kapitole 5. Volání funkce:

```
1 GetGoogleSpreadsheet(spreadsheetID,Ranges,Fields,Key)
```

Využijeme stejné parametry a tabulku jako v kapitole 5.1.3. Výsledek:

```
1 {
2 "sheets": [{
3   "data": [{
4     "rowData": [{
5       "values": [{
6         "formattedValue": "VšB 2325",
7         "userEnteredFormat": {
8           "numberFormat": {
9             "type": "NUMBER",
10            "pattern": "#,##0.00;(#,##0.00)",
11            "backgroundColor": {"red": 0.8},
12            "borders": {
13              "top": {
14                "style": "SOLID_THICK",
15                "width": 3,
16                "color": {"blue": 1}},
17              "bottom": {
18                "style": "SOLID_THICK",
19                "width": 3,
20                "color": {"blue": 1}},
21              "left": {
22                "style": "SOLID_THICK",
23                "width": 3,
24                "color": {"blue": 1}},
25              "right": {
26                "style": "SOLID_THICK",
27                "width": 3,
28                "color": {"blue": 1}},
29              "horizontalAlignment": "CENTER",
30              "verticalAlignment": "MIDDLE",
31              "wrapStrategy": "WRAP",
32              "textFormat": {
33                "foregroundColor":
34                  {"red": 1,"green": 1},
35                "fontFamily": "Lobster",
36                "fontSize": 18,
37                "bold": true,
38                "italic": true },
39              "textRotation": {
40                "vertical": false}},
41            "effectiveFormat": {
42              "textRotation": {
43                "angle": -90}}}}],
44      "rowMetadata": [{
45        "pixelSize": 121}],
46      "columnMetadata": [{
47        "pixelSize": 155},{
48        "pixelSize": 166},{
49        "pixelSize": 100}}],
50      "merges": [{
51        "sheetId": 1489130067,
52        "startRowIndex": 0,"endRowIndex": 1,"startColumnIndex": 1,"endColumnIndex": 3}]}}
```

Zde lze vidět, že výpis je stejný jako v kapitole 5.1.3

8.3 Zápis hodnot

Následuje stejný postup jako při zápisu hodnot z Matlabu. To znamená vytvoření Client ID, Client Secret a vygenerování přístupových tokenů, viz 6.2. Až poté je možno zapisovat. Princip zápisu je popsán v kapitole 6.3

Pro vygenerování přístupových tokenů použijeme RunOnce.sci, nacházející se ve složce /Scilab/Scilab2sheets v příloze, stejně tak program pro zápis sci2sheets.sci.

Mat2sheets.m a RunOnce.sci vycházejí z programů, jež jsem využíval v Matlabu, viz 6.3. Volání funkce:

```
1 sci2sheets(spreadsheetID, sheetName, sheetpos, d)
```

Rozdíl nastává při zapisování různých datových typů najednou. Narozdíl od Matlabu, Scilab není schopen přijmout pole, ve kterém se nachází číselné datové typy a string. Zápis double a int možný je. Pokud tedy chceme zapsat string a čísla, znamená to, že číselné hodnoty musíme napsat jako string. Pro příklad budeme volat funkci se stejnými hodnotami jako v kapitole 6.3.1

```
1 spreadsheetID = '11EOPaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetName = 'ListHodnoty2';
3
4 sci2sheets(spreadsheetID, sheetName, [2 2], ['INT' 'DOUBLE' 'SYMBOL' 'STRING'; '1'
    '1.1' 'a' 'ab'; '2' '2.22' 'b' 'cd'; '3' '3.333' 'c' 'ef'; '5' '5.5555' 'd' '
    gh'; '7' '7.7777' 'e' 'ch'; '11' '11.1111' 'f' 'ij'; '13' '13.1313' 'g' 'kl'; '
    17' '17.1717' 'h' 'mn'; '19' '19.1919' 'i' 'op'])
```

	A	B	C	D	E
1					
2		INT	DOUBLE	SYMBOL	STRING
3		1	1.1	a	ab
4		2	2.22	b	cd
5		3	3.333	c	ef
6		5	5.5555	d	gh
7		7	7.7777	e	ch
8		11	11.1111	f	ij
9		13	13.1313	g	kl
10		17	17.1717	h	mn
11		19	19.1919	i	op
12					

Obrázek 36: Zápis hodnot Scilab

Zapsané hodnoty budou ve Scilabu zapsané do Google Sheets vždy jako string, to znamená, že budou zarovnány vlevo. Pokud zapíšeme číselné hodnoty přes Matlab, budou zarovnány vpravo, viz 6.3.1.

8.4 Změna barvy pozadí

Změnu barvy a všechny ostatní editace tabulek budeme provádět v programu `sciProperties2sheets.sci`. Ten najdeme v příloze ve složce `/Scilab/ScilabProperties2sheets`.

Změna barvy pozadí je popsána v kapitole 6.4. Ve Scilabu probíhá editace barvy pozadí stejně. Pro tuto a všechny ostatní akce budeme volat funkci:

```
1 sciProperties2sheets(spreadsheetID, sheetID, D, p)
```

Pro všechny následující ukázky budu používat stejné vlastnosti jako při ukázkách v Matlabu. Parametry (`spreadsheetID` a `sheetID`) budeme využívat ve všech dalších editacích tabulky, takže je dál nebudu zmiňovat. Detailní popis funkcí lze vyčíst z ukázek v Matlabu.

Ukázka:

```
1 spreadsheetID = '11EOPaV01w_e8wopo1cZLIAiDM06TV3ICvuL8hUeWYws';
2 sheetID = '833404061';
3
4 sciProperties2sheets(spreadsheetID, sheetID, {'BarvaPozadi'}, [2 2 6 3 255 255
    125 ])
```

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1.1	a	ab	
4		2	2.22	b	cd	
5		3	3.333	c	ef	
6		5	5.5555	d	gh	
7		7	7.7777	e	ch	
8		11	11.1111	f	ij	
9		13	13.1313	g	kl	
10		17	17.1717	h	mn	
11		19	19.1919	i	op	
12						

Obrázek 37: Změna barvy pozadí Scilab

8.5 Změna barvy textu

Editace barvy textu je popsána v kapitole 6.5.

Ukázka:

```
1 sciProperties2sheets(spreadsheetID, sheetID,{'BarvaTextu'},[2 2 6 3 250 0 0 ])
```

	A	B	C	D	E	F
1						
2		INT	DOUBLE	SYMBOL	STRING	
3		1	1.1	a	ab	
4		2	2.22	b	cd	
5		3	3.333	c	ef	
6		5	5.5555	d	gh	
7		7	7.7777	e	ch	
8		11	11.1111	f	ij	
9		13	13.1313	g	kl	
10		17	17.1717	h	mn	
11		19	19.1919	i	op	
12						
13						

Obrázek 38: Změna barvy textu Scilab

8.6 Změna formátu

Změna formátu je vysvětlena v kapitole 6.6. Rozdíl nastává v zadávání vlastností. Jak jsem již zmiňoval, Scilab nedokáže přijímat pole s numerickým datovým typem a stringem. V zadávaných vlastnostech pro změnu formátu se nachází string (jméno fontu). V tomto případě je nutno všechny ostatní vlastnosti v tomto poli zadávat jako string.

Ukázka:

```
1 sciProperties2sheets(spreadsheetID, sheetID, {'FormatTextu'}, ['2' '2' '6' '3' '15  
  ' 'Courier new' '1' '1' '0' '1'])  
2  
3 sciProperties2sheets(spreadsheetID, sheetID, {'FormatTextu'}, ['2' '3' '6' '12' '  
  11' 'Georgia' '0' '1' '0' '0'])
```

	A	B	C	D	E	F
1						
2		<i>INT</i>	<i>DOUBLE</i>	<i>SYMBOL</i>	<i>STRING</i>	
3		<i>1</i>	<i>1.1</i>	<i>a</i>	<i>ab</i>	
4		<i>2</i>	<i>2.22</i>	<i>b</i>	<i>cd</i>	
5		<i>3</i>	<i>3.333</i>	<i>c</i>	<i>ef</i>	
6		<i>5</i>	<i>5.5555</i>	<i>d</i>	<i>gh</i>	
7		<i>7</i>	<i>7.7777</i>	<i>e</i>	<i>ch</i>	
8		<i>11</i>	<i>11.1111</i>	<i>f</i>	<i>ij</i>	
9		<i>13</i>	<i>13.1313</i>	<i>g</i>	<i>kl</i>	
10		<i>17</i>	<i>17.1717</i>	<i>h</i>	<i>mn</i>	
11		<i>19</i>	<i>19.1919</i>	<i>i</i>	<i>op</i>	
12						

Obrázek 39: Editace formátu Scilab

8.7 Editace ohraničení

Popis editace ohraničení najdeme v kapitole 6.7. Tak jako v předchozím případě, vlasnosti v poli 'p' musíme zadávat jako stringy. Ukázka:

```
1 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '2' '6' '12' '
   top' 'SOLID' '3' '0' '0' '255'])
2 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '2' '6' '12' '
   right' 'SOLID' '3' '0' '0' '255'])
3 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '2' '6' '12' '
   bottom' 'SOLID' '3' '0' '0' '255'])
4 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '2' '6' '12' '
   left' 'SOLID' '3' '0' '0' '255'])
5 //Ohraničení dokola tabulky
6 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '2' '6' '3' '
   bottom' 'SOLID' '3' '0' '0' '255'])
7 //Spodní ohraničení záhlaví
8 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '3' '6' '12' '
   innerHorizontal' 'DASHED' '1' '255' '100' '0'])
9 sciProperties2sheets(spreadsheetID,sheetID,{'Ohranicieni'},['2' '3' '6' '12' '
   innerVertical' 'DOTTED' '1' '0' '255' '255'])
10 //Vertikální a horizontální ohraničení uvnitř tabulky
```

Výsledná tabulka:

	A	B	C	D	E	F
1						
2		<i>INT</i>	<i>DOUBLE</i>	<i>SYMBOL</i>	<i>STRING</i>	
3		<i>1</i>	<i>1.1</i>	<i>a</i>	<i>ab</i>	
4		<i>2</i>	<i>2.22</i>	<i>b</i>	<i>cd</i>	
5		<i>3</i>	<i>3.333</i>	<i>c</i>	<i>ef</i>	
6		<i>5</i>	<i>5.5555</i>	<i>d</i>	<i>gh</i>	
7		<i>7</i>	<i>7.7777</i>	<i>e</i>	<i>ch</i>	
8		<i>11</i>	<i>11.1111</i>	<i>f</i>	<i>ij</i>	
9		<i>13</i>	<i>13.1313</i>	<i>g</i>	<i>kl</i>	
10		<i>17</i>	<i>17.1717</i>	<i>h</i>	<i>mn</i>	
11		<i>19</i>	<i>19.1919</i>	<i>i</i>	<i>op</i>	
12						

Obrázek 40: Editace ohraničení Scilab

8.8 Přidávání grafů

Styl přidávání grafů vyčteme v kapitole 6.8. Stejně jako v předchzích dvou kapitolách, vlastnosti zadáváme jako stringy.

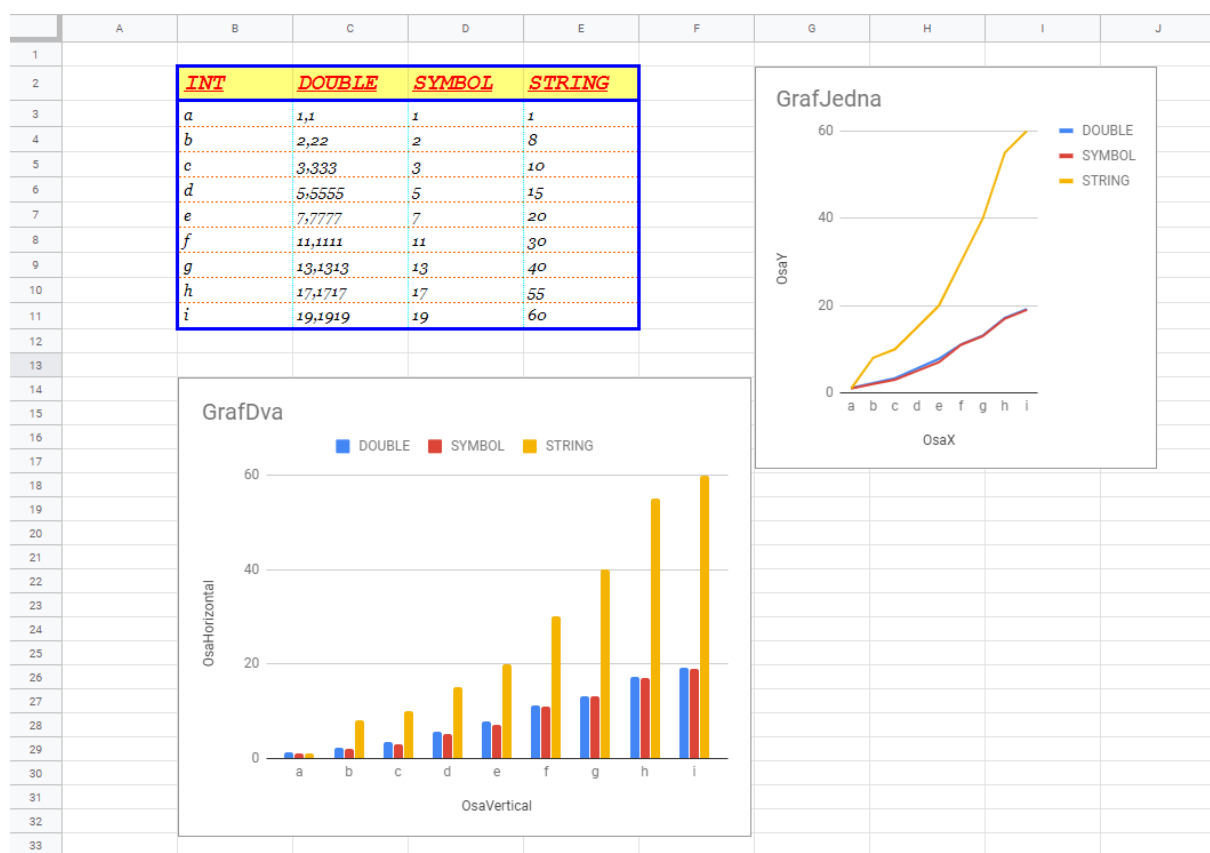
Pro ukázky s grafy jsem tabulku, z níž budeme číst data, upravil stejným způsobem jako na obrázku 33.

Ukázka:

```

1 sciProperties2sheets(spreadsheetID, sheetID,{'Grafy'}, ['2' '2' '6' '12' 'GrafJedna' '
   LINE' 'Right_LEGEND' 'OsaX' 'OsaY' '2' '7' '350' '350'])
2 sciProperties2sheets(spreadsheetID, sheetID,{'Grafy'},['2' '2' '6' '12' 'GrafDva' '
   COLUMN' 'TOP_LEGEND' 'OsaVertical' 'OsaHorizontal' '14' '2' '500' '400'])

```



Obrázek 41: Přidání grafů Scilab

Na pravé straně vidíme graf vygenerovaný první funkcí. Druhá funkce vygenerovala graf na spodní straně.

8.9 Práce s listy

Práce s listy je stejná ve Scilabu jako v Matlabu, viz 7 .

Ukázky:

- přidání listu

```
1 sciProperties2sheets(spreadsheetID, [],{'PridatList'}, {'ListJedna'})
```

- přejmenování listu

```
1 sheetID = '469972044';  
2 sciProperties2sheets(spreadsheetID, sheetID,{'PrejmenovatList'},{'ListNovy'})
```

- zkopírování listu

```
1 sheetID = '469972044';  
2 sciProperties2sheets(spreadsheetID, sheetID,{'KopirovatList'})
```

- smazání listu

```
1 sheetID = '1893510200';  
2 sciProperties2sheets(spreadsheetID, sheetID,{'SmazatList'})
```

9 Zhodnocení

Cílem této práce bylo pospat základní a pokročilé možnosti práce s daty v Google sheets z prostředí MATLAB a Scilab.

Práce je rozdělená na 4 části, synchronizaci dat mezi lokálním uložištěm a uložištěm Google Drive, výpis hodnot a vlastností buněk a tabulky do prostředí MATLAB, zápis dat do Google Sheets z prostředí MATLAB a práci z Google Sheets prostřednictvím Scilabu.

V kapitole synchronizace dat popisují možnosti synchronizace dat na platformách Windows 10 a Linux Ubuntu 18.04. U Windowsu se zaměřuji na program Backup and Sync. U Linuxu popisují Gnome Online Accounts a Google Drive Ocamlfuse.

Další část obsahuje popis, jakým způsobem lze vyčítat jak samotné hodnoty, tak vlastnosti a hodnoty zároveň. O tom, zda-li chceme vyčítat jen hodnoty a nebo i vlastnosti, rozhoduje, jak zadáme URL a poté parametr 'Fields', který určuje, jaké vlastnosti chceme vypsat.

Co se týče importu dat, zde se zabývám základními možnostmi, jako je zápis hodnot, ale také pokročilejšími možnostmi, což je změna barvy pozadí a textu, editace ohraničení, editace formátu textu a přidávání grafů. Poté také zhodnocuji možnosti práce s listy.

Jako poslední se zabývám možnostmi práce s Google Sheets prostřednictvím Scilabu. Práce se Scilabem je dost podobná té v Matlabu, proto jsou zde vypsány hlavní rozdíly mezi prací v Scilabu a Matlabu.

Práce s Google Sheets z prostředí MATLAB a Scilab v off-line režimu není možná. Jediná možnost jak editovat tabulky v Google Sheets off-line je přes prohlížeč Google Chrome nebo Google Drive, které jsou přepnuté do offline režimu. Po následném připojení k internetu jsou změny, které byly provedeny v off-line režimu, nahrány online.

Při vytváření programů pro import a export dat na platformě Windows 10 jsem nepostřehl žádný problém. V prostředí MATLAB na platformě Linux Ubuntu 18.04 import a export dat do Google Sheets probíhá bez problému, rozdíl je akorát v znakové sadě, diakritika se nezobrazuje správně. Instalace Scilabu 5.5.2 na Linux Ubuntu 18.04 se mi nezdařila, jelikož tato verze Scilabu není kompatibilní z Linux Ubuntu 18.04. Platformu MAC jsem neotestoval, protože k ní nemám přístup.

Literatura

- [1] Yair M. Altman. Undocumented Secrets of MATLAB-Java Programming. Boca Raton: CRC Press, 2011. ISBN 978-1-4398-6903-
- [2] Yair M. Altman. Accelerating MATLAB Performance: 1001 tips to speed up MATLAB programs. Boca Raton: CRC Press, 2014. ISBN 978-1-4822-1130-6
- [3] Introduction to the Google Sheets API. Common terms. [online].[cit. 2019-01-10]. Dostupné z: <https://developers.google.com/sheets/api/guides/concepts/>
- [4] Daniel. GetGoogleSpreadsheet. File Exchange. [online]. [cit. 2019-02-03]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/39915-getgooglespreadsheet>
- [5] Andrew Bogaard. Matlab to Google Sheets (matlab2sheets). File Exchange. [online]. [cit. 2019-01-10]. Dostupné z: <https://www.mathworks.com/matlabcentral/fileexchange/59359-matlab-to-google-sheets-matlab2sheets>

Seznam příloh

- GetGoogleSpreadsheet.m - Příloha na CD /GoogleSheetsToMatlab/GetGoogleSpreadsheet.m
- GoogleSheetsValuesToMatlab.m - Příloha na CD /GoogleSheetsValuesToMatlab/GoogleSheetsValuesToMatlab.m
- RunOnce.m - Příloha na CD /Matlab2sheets/RunOnce.m
- mat2sheets.m - Příloha na CD /Matlab2sheets/mat2sheets.m
- matProperties2sheets.m - Příloha na CD /MatlabProperties2sheets/matProperties2sheets.m
- GetGoogleSpreadsheet.sci - Příloha na CD /Scilab/GetGoogleSpreadsheet.sci
- GoogleSheetsValuesToScilab.sci - Příloha na CD /Scilab/GoogleSheetsValuesToScilab.sci
- sci2sheets.sci - Příloha na CD /Scilab/Scilab2sheets/sci2sheets.sci
- RunOnce.sci - Příloha na CD /Scilab/Scilab2sheets/RunOnce.sci
- sciProperties2sheets.sci - Příloha na CD /Scilab/ScilabProperties2sheets/sciProperties2sheets.sci